

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Moderné metódy riešenia konvexných optimalizačných úloh
DIZERTAČNÁ PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



Moderné metódy riešenia konvexných optimalizačných úloh

DIZERTAČNÁ PRÁCA

Študijný program: Aplikovaná matematika

Študijný odbor: Matematika

Školiace pracovisko: FMFI KAMŠ - Katedra aplikovanej matematiky a štatistiky

Školiteľ: doc. RNDr. Margaréta Halická, CSc.

Konzultant: doc. RNDr. Mária Trnovská, PhD.

BRATISLAVA 2021

Mgr. Roman Kukumberg



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Mgr. Roman Kukumberg
Študijný program: aplikovaná matematika (Jednoodborové štúdium, doktorandské III. st., denná forma)
Študijný odbor: matematika
Typ záverečnej práce: dizertačná
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Moderné metódy riešenia konvexných optimalizačných úloh
Modern methods for solving convex optimization problems

Anotácia: Práca sa zaoberá modernými metódami konvexnej optimalizácie vhodnými na riešenie nediferencovateľných úloh špeciálnej štruktúry. Konkrétne je práca zameraná na tzv. l_1 -regulované úlohy, ktoré obsahujú vo svojej účelovej funkcii l_1 normu. Práca poskytne prehľad známych metód, navrhne ich prípadné modifikácie a numerické vylepšenia. Otestuje a porovná výkon metód na vygenerovaných dátach. Aplikuje vybrané metódy na reálne problémy.

Cieľ: Témou dizertačnej práce sú moderné metódy konvexnej optimalizácie pre veľkorozmerné úlohy na voľný extrém, kde účelová funkcia má špeciálnu štruktúru a je vo všeobecnosti nediferencovateľná. Cieľom práce je spracovať prehľad známych metód, ich teoretická analýza, navrhnutie prípadných modifikácií a numerických vylepšení, testovanie a porovnanie jednotlivých metód vykonaním numerických experimentov na vygenerovaných dátach a aplikácia vybraných metód na reálne problémy z rôznych oblastí.

Školiteľ: doc. RNDr. Margaréta Halická, CSc.
Konzultant: doc. RNDr. Mária Trnovská, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Marek Fila, DrSc.
Dátum zadania: 21.03.2013

Dátum schválenia: 21.03.2013
prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
školiteľ

Podakovanie

Aj touto cestou by som sa ešte rád poďakoval svojej školiteľke doc. RNDr. Margaréte Halickej, CSc. a takisto svojej konzulantke doc. RNDr. Márii Trnovskej, PhD. za odborné vedenie, cenné rady, ochotu a čas pri tvorení tejto práce. Takisto by som sa rád poďakoval aj všetkým ostatným, ktorí ma pri tvorbe tejto práce podporovali. Predovšetkým ďakujem mojej milujúcej rodine ako aj najlepšiemu kamarátovi Radovanovi Činčurovi, ktorého podpora a hudba ma sprevádzala pri písaní tejto práce. Taktiež by som sa rád poďakoval všetkým ľuďom z katedry aplikovanej matematiky a štatistiky za prijatie a priateľskú atmosféru v pracovnom prostredí.

Abstrakt

Mgr. Roman Kukumberg: *Moderné metódy riešenia konvexných optimalizačných úloh* [Dizertačná práca]; Univerzita Komenského v Bratislave; Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej matematiky a štatistiky; školiteľ: doc. RNDr. Margaréta Halická, CSc.; Bratislava; 2021.

Práca sa zaoberá modernými metódami riešenia konvexných optimalizačných úloh, presnejšie veľkorozmerných úloh na voľný extrém so špeciálnou štruktúrou, kde účelová funkcia je rozložiteľná na konvexnú diferencovateľnú a nediferencovateľnú časť. Venujeme sa hlavne tzv. l_1 -regulovaným úlohám, ktoré majú v praxi široké uplatnenie. Medzi najvýznamnejších predstaviteľov týchto úloh patrí úloha l_1 -regulovaných najmenších štvorcov a úloha l_1 -regulovanej logistickej regresie, pri ktorých sme urobili prehľad aplikácií a metód riešenia. Medzi moderné metódy riešenia patria proximálne metódy a metódy vnútorného bodu, ktoré podrobnejšie skúmame, navrhujeme ich modifikácie a numerické vylepšenia. Navrhujeme hybridnú metódu, ktorá kombinuje algoritmus primárno-duálnych metód vnútorného bodu s poznatkami z proximal-gradientných metód. Takisto sa venujeme spôsobu implementácie metód na riešenie l_1 -regulovaných úloh. Vykonávame rozsiahle numerické experimenty a porovnanie metód na vygenerovaných dátach za rôznych podmienok. Testujeme napríklad vplyv veľkosti rozmerov a riedkosti dát, či vplyv hodnoty regularizačného parametra λ na výkon metód. Takisto vybrané metódy aplikujeme na riešenie reálnych problémov z rôznych oblastí ako optimálny topologický dizajn, spracovanie signálov, medicína, kategorizácia textu.

Kľúčové slová: proximálny operátor, proximal-gradientná metóda, primárno-duálna metóda vnútorného bodu, l_1 -regulované úlohy, l_1 -regulované najmenšie štvorce, l_1 -regulovaná logistická regresia, dizajn mechanickej konštrukcie, rekonštrukcia riedkeho signálu

Abstract

Mgr. Roman Kukumberg: *Modern methods for solving convex optimization problems* [Dissertation thesis]; Comenius University in Bratislava; Faculty of mathematics, physics and informatics; Department of applied mathematics and statistics; supervisor: doc. RNDr. Margaréta Halická, CSc.; Bratislava; 2021.

The dissertation thesis deals with modern methods for solving convex optimization problems, more precisely large-scale unconstrained optimization problems with a special structure, where the objective function can be split into a convex differentiable and non-differentiable part. We mainly deal with the so-called l_1 -regularized problems that have a wide application in practice. The most important representatives of these problems are the l_1 -regularized least squares and the l_1 -regularized logistic regression, in which we made an overview of applications and solution methods. Modern methods include proximal methods and interior-point methods, which we examine in more detail, propose modifications and numerical improvements. We propose a hybrid method that combines the algorithm of the primal-dual interior-point method with the proximal-gradient method. We also discuss how to implement methods for solving l_1 -regularized problems. We perform extensive numerical experiments and comparison of methods on generated data under different conditions. For example, we test the effect of problem size, data sparsity, or the effect of the regularization parameter λ on the performance of methods. We also apply selected methods to solve real problems from various areas such as optimal topology design, signal processing, medicine, text categorization.

Keywords: proximal operator, proximal-gradient method, primal-dual interior point method, l_1 -regularized problems, l_1 -regularized least squares, l_1 -regularized logistic regression, truss topology design, sparse signal reconstruction

Obsah

Zoznam použitých symbolov a skratiek	1
Úvod	3
1 Úlohy so špeciálnou štruktúrou	10
1.1 Štruktúra úlohy (1.1) z pohľadu f	11
1.2 Štruktúra úlohy (1.1) z pohľadu g	12
1.3 Úloha l_1 -regulovaných najmenších štvorcov	14
1.3.1 Vlastnosti úlohy l_1 -regulovaných najmenších štvorcov	14
1.3.2 Prehľad aplikácií a metód riešenia úlohy l_1 -regulovaných najmen- ších štvorcov	15
1.4 Úloha l_1 -regulovanej logistickej regresie	17
1.4.1 Úloha logistickej regresie	17
1.4.2 Formulácia a vlastnosti úlohy l_1 -regulovanej logistickej regresie .	19
1.4.3 Prehľad aplikácií a metód riešenia úlohy l_1 -regulovanej logistickej regresie	20
I Metódy riešenia	22
2 Proximálne metódy	24
2.1 Proximálny operátor	24
2.2 Proximal-gradientná metóda (PGM)	27
2.2.1 Formulácia a základné vlastnosti PGM	27
2.2.2 Interpretácie PGM	29

2.2.3	Zrýchlená proximal-gradientná metóda (APGM)	32
2.2.4	Spádové modifikácie APGM	34
2.3	Proximal-newtonova metóda (PNM)	36
2.3.1	Formulácia a základné vlastnosti PNM	36
2.3.2	Interpretácie PNM	37
2.4	Nová proximal-kvázinewtonova metóda (PQNM)	40
2.4.1	Motivácia a popis PQNM	40
2.4.2	Interpretácie PQNM	41
2.5	Prírastková proximal-gradientná metóda (PPGM)	45
2.5.1	Popis a základné vlastnosti PPGM	45
2.5.2	Modifikácie prírastkovej proximal-gradientnej metódy	47
2.6	ADMM metóda	48
2.6.1	Popis a základné vlastnosti ADMM	49
2.6.2	Iterpretácia ADMM metódy	50
3	Metódy vnútorného bodu	51
3.1	Formulácia úlohy a základná myšlienka MVB	51
3.2	Primárno-duálne metódy vnútorného bodu	52
3.2.1	Primárno-duálny smer v MVB	53
3.2.2	Primárno-duálny algoritmus metódy vnútorného bodu	55
3.3	Nová PDMVB s redukciou dimenzie	57
3.3.1	Algoritmus PDMVB s redukciou dimenzie	59
4	Implementácia metód riešenia	63
4.1	Odvodenie a notácia základných stavebných prvkov algoritmov	63
4.2	Implementácia proximálnych metód	65
4.2.1	Proximálne operátory	65
4.2.2	Voľba dĺžky kroku	68
4.3	Implementácia metód vnútorného bodu	73
4.3.1	Prvý spôsob transformácie l_1 -regulovanej úlohy	74
4.3.2	Druhý spôsob transformácie l_1 -regulovanej úlohy	74
4.3.3	Primárno-duálny smer pre (4.34) a (4.35)	75

4.3.4	Riešenie Newtonovych systémov lineárnych rovníc	79
II	Numerické experimenty a aplikácie	86
5	Metodika testovania a značenie metód	87
6	Numerické experimenty	92
6.1	Metódy riešenia	92
6.1.1	Proximálne metódy	93
6.1.2	Prírastkové PG metódy	95
6.1.3	PDMVB	98
6.2	Úloha l_1 -regulovaných najmenších štvorcov	102
6.2.1	Kvalifikácia metód	103
6.2.2	Globálne pozorovania z tabuliek	106
6.2.3	Vplyv regularizačného parametra λ	108
6.2.4	Vplyv rozmeru m	111
6.2.5	Vplyv rozmeru n	113
6.2.6	Vplyv rozmerov $m \times n$	115
6.2.7	Vplyv riedkosti matice dát 1	117
6.2.8	Vplyv riedkosti matice dát 2	120
6.2.9	Vplyv vzdialenosti od optimálneho riešenia	124
6.2.10	Vplyv počtu nenulových prvkov optimálneho riešenia	127
6.2.11	Vplyv dynamického rozsahu optimálneho riešenia	130
6.2.12	Škálovateľnosť	132
6.3	Úloha l_1 -regulovanej logistickej regresie	137
6.3.1	Kvalifikácia metód	138
6.3.2	Vplyv regularizačného parametra λ	140
6.3.3	Vplyv rozmeru m	143
6.3.4	Vplyv rozmeru n	145
6.3.5	Vplyv rozmerov $m \times n$	147
6.3.6	Vplyv riedkosti matice dát	149
6.3.7	Škálovateľnosť	152

7 Aplikácie	158
7.1 Dizajn mechanickej konštrukcie	159
7.1.1 Porovnanie metód	160
7.2 Rekonštrukcia riedkeho signálu	172
7.2.1 Porovnanie metód	172
7.3 l_1 -regulovaná logistická regresia	182
7.3.1 Voľba hodnoty regularizačného parametra λ	183
7.3.2 Porovnanie metód	188
Záver	198
Literatúra	203
A Prílohy	214
A.1 Vlastnosti funkcií	214
A.2 Subgradient funkcie	215
A.3 Rýchlosť konverencie	216
A.4 Sherman–Morrison–Woodbury formula	216
B Tabuľky	217

Zoznam použitých symbolov a skratiek

$\mathbb{R}^n$	priestor reálnych n -rozmerných vektorov
$\nabla f(x)$	gradient funkcie $f(x)$, $\nabla f(x) = (\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n})^T$
$\nabla^2 f(x)$	Hessova matica funkcie $f(x)$
$\partial g(x)$	subdiferenciál funkcie g v bode x
$Df(x)$	$m \times n$ Jakobiho matica funkcie $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
F^*	optimálna hodnota funkcie $F(x)$
x^*	bod minima funkcie $F(x)$
x_0	štartovací bod
x_k	bod iterácie k
t_k	dĺžka kroku v iterácii k
L	konštanta z Lipschitzovskej vlastnosti gradientu
$\ \cdot\ _0$	l_0 „norma“, $\ x\ _0 = \#(i x_i \neq 0)$
$\ \cdot\ _1$	l_1 /Manhattanská norma, $\ x\ _1 = \sum_{i=1}^n x_i $
$\ \cdot\ _2$	Euklidovská norma, $\ x\ _2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^n x_i^2}$
$\ \cdot\ _H$	škálovaná norma, $\ x\ _H = \sqrt{x^T H x}$, $H \succ 0$
$prox_g(\cdot)$	proximálny operátor funkcie g , $prox_g(x) = \operatorname{argmin}_u (g(u) + \frac{1}{2}\ u - x\ _2^2)$
$prox_g^H(\cdot)$	škálovaný prox. op. funkcie g , $prox_g^H(x) = \operatorname{argmin}_u (g(u) + \frac{1}{2}\ u - x\ _H^2)$
$I_C(\cdot)$	Indikátorová funkcia, $I_C(x) = \begin{cases} 0 & x \in C \\ \infty & \text{inak} \end{cases}$
$P_C(\cdot)$	projekcia na množinu C , $P_C(x) = \operatorname{argmin}_{u \in C} \ u - x\ _2^2$
$p(b a)$	podmienená pravdepodobnosť udalosti b za udalosti a

$diag(x)$	diagonálna matica so zložkami vektora x na diagonále
$diag(A)$	vektor diagonálnych prvkov matice A
$L(\cdot)$	Lagrangeova funkcia
$\mathbf{1}$	vektor samých jednotiek, t.j. $\mathbf{1} = (1, \dots, 1)^T$
$(\cdot)_+$	kladná časť, $(x)_+ = \max(0, x)$
$sgn(\cdot)$	signum funkcia, $sgn(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$
PGM	proximal-gradientná metóda
APGM	zrýchlená proximal-gradientná metóda
FISTA	Fast Iterative Shrinkage Thresholding Algorithm
PNM	proximal-newtonova metóda
PQNM	proximal-kvázinewtonova metóda
PPGM	prírastková proximal-gradientná metóda
ADMM	Alternating Direction Method of Multipliers
IAG	Incremental Aggregated Gradient
SAG	Stochastic Average Gradient
PIAG	proximálna verzia IAG
MISO	Minimization by Incremental Surrogate Optimization
BBR	Bayesian Binary Regression
LARS	Least-Angle Regression
SUMT	Sequential Unconstrained Minimization Technique
MVB	metódy vnútorného bodu
PDMVB	primárno-duálne metódy vnútorného bodu
KKT	Karush-Kuhn-Tucker
LASSO	Least Absolute Shrinkage and Selection Operator
TTD	Truss Topology Design / dizajn mechanickej konštrukcie
SSR	Sparse Signal Reconstruction / rekonštrukcia riedkeho signálu

Úvod

Matematické programovanie ako matematická disciplína, ktorá skúma vlastnosti a navrhuje metódy riešenia optimalizačných úloh, prešla v uplynulých 40 rokoch veľkými zmenami a vývojom. Zmeny boli vyvolané rozvojom výpočtovej techniky a možnosťami výkonnej techniky riešiť úlohy veľkých rozmerov. V popredí záujmu začali byť vo významnej miere otázky výpočtovej zložitosti a efektívnosti algoritmov. Ako výsledok tohto vývoja nadobudli na význame niektoré triedy konvexných optimalizačných úloh, pre ktoré boli vyvinuté efektívne metódy a algoritmy umožňujúce riešiť takéto úlohy viac-menej rutinne.

Konvexné optimalizačné úlohy

Pod konvexnou optimalizačnou úlohou alebo úlohou konvexného programovania rozumieme takú úlohu matematického programovania, v ktorej hľadáme minimum konvexnej funkcie na konvexnej množine. Dôležitou vlastnosťou je, že každé lokálne minimum konvexnej optimalizačnej úlohy je zároveň aj jej globálnym minimom. Práve táto vlastnosť konvexných úloh umožňuje konštrukciu algoritmov, pomocou ktorých možno efektívne riešiť aj veľké úlohy. Hoci predpoklad konvexnosti je pomerne silným globálnym predpokladom, až prekvapivo veľa problémov z praxe vedie na takéto úlohy. Medzi aplikácie konvexných optimalizačných úloh patria inžinierske, finančné aplikácie (napr. optimalizácia portfólia), aplikácie vyskytujúce sa pri kompresii a analýze dát, či pri rôznych geometrických úlohách. Úlohy konvexnej optimalizácie možno nájsť taktiež v optimálnom riadení, či v topologickom a v experimentálnom optimálnom dizajne. Takisto niektoré nekonvexné úlohy sa často riešia pomocou konvexnej relaxácie úlohy alebo pomocou algoritmov založených na riešení konvexných podúloh.

Špeciálnym typom konvexných optimalizačných úloh sú úlohy, v ktorých sa hľadá minimum konvexnej funkcie bez akýchkoľvek ďalších ohraničení. Takéto úlohy nazývame konvexnými úlohami na voľný extrém, resp. úlohami voľnej konvexnej minimalizácie. Aj takéto úlohy majú v praxi široké uplatnenie, a ich význam neustále rastie. V dôsledku digitalizácie a informatizácie vznikajú potreby spracovávať veľmi rozsiahle dátové súbory. Niektoré spôsoby spracovania vedú na úlohy voľnej konvexnej optimalizácie veľkých rozmerov. Napríklad v strojovom učení (angl. machine-learning) a v štatistike pri aproximácii a fitovaní dát je to úloha najmenších štvorcov a úloha nelineárnych najmenších štvorcov. Úlohy na voľný extrém sa takisto používajú na klasifikáciu dát (napr. úloha logistickej regresie). Rôzne typy účelových funkcií používaných v štatistike sú spracované v [96], kde autori zhromaždili zoznam konvexných funkcií merajúcich chybu pri fitovaní dát. Úlohy na voľný extrém možno nájsť taktiež pri spracovaní, rekonštrukcii, odšumovaní či vyhladzovaní signálu (či už jednorozmerného zvukového alebo viacrozmerného ako obrázky a video) alebo aj pri zaostrovaní obrázkov či kompresii dát. S ďalšími aplikáciami týchto úloh sa možno stretnúť aj pri optimálnom rozmiestňovaní objektov a pri úlohách optimálneho dizajnu (napr. dizajn nosnej konštrukcie). Úlohy na voľný extrém sa dajú použiť aj na analýzu dát, v ekonómii sa využívajú pri hľadaní podkladového trendu v dátach (pri odstraňovaní sezónnosti - Hodrick-Prescott filter, l_1 filter) alebo pri hľadaní štrukturálnych zlomov. Úlohy na voľný extrém sa takisto používajú ako pomocné podúlohy pri riešení úloh s ohraničeniami.

Metódy riešenia konvexných optimalizačných úloh

Medzi klasické metódy pre hľadanie globálneho minima konvexnej funkcie patrí gradientná metóda a Newtonova metóda. Jednou z prvých a najznámejších metód gradientného typu je Cauchyho metóda najväčšieho spádu (gradientná metóda s voľbou optimálnej dĺžky kroku), ktorú navrhol Cauchy v roku 1847 [23]. Newtonova metóda je známa svojou lokálne kvadratickou rýchlosťou konverencie¹, vďaka čomu vie vyriešiť úlohy na voľný extrém za desiatky iterácii. Avšak pre úlohy väčších rozmerov môže byť častokrát príliš náročná, aj nutnosť počítania Hessovej matice robí jednotlivé iterácie Newtonovej metódy náročnými. Naopak, gradientné metódy majú jednoduchšiu, vý-

¹Pozri Príloha A.3.

počtovo nenáročnú iteráciu a menšie pamäťové nároky a z týchto dôvodov môžu byť pri veľkých rozmeroch úlohy konkurencieschopné. Medzi ďalšie metódy riešenia patria aj tzv. Kvázinewtonovské metódy, ktoré skĺbujú dobré vlastnosti gradientných metód a Newtonovej metódy. Tieto metódy pomocou gradientnej informácie postupne aproximujú inverziu Hesseovej matice, a preto pri nich nie je potrebné riešiť systém lineárnych rovníc. Gradientnú a Newtonovu metódu a k nim odvodené konvergenčné vlastnosti či rôzne modifikácie možno nájsť napríklad v [48], [14], [79], [9], [84], [3], kvázinewtonovské metódy sú spracované v [48], [9].

V prípade konvexných úloh s ohraničeniami, kde ohraničenia definujú konvexnú množinu prípustných riešení, sú v súčasnosti najpoužívanějšími metódami riešenia metódy vnútorného bodu (MVB). História týchto metód siaha až do 60-tych rokov 20. storočia, kedy Fiacco a McCormick vytvorili metódu postupného riešenia úloh na voľný extrém - tzv. SUMT (Sequential Unconstrained Minimization Technique) nazývanú dnes aj ako bariérová metóda vnútorného bodu [33] [34]. Metódy vnútorného bodu boli však v tom období aplikované na zložité nelineárne úlohy a príliš sa neuchytili. Znovuobnovenie záujmu o metódy vnútorného bodu nastalo v 80-tych rokoch, keď Karmarkar navrhol na riešenie úloh lineárneho programovania tzv. projektívny algoritmus [53], ktorý súvisel s bariérovými metódami vnútorného bodu a mal polynomiálnu zložitosť.

Základnou myšlienkou bariérových metód vnútorného bodu je riešiť pôvodnú úlohu s ohraničeniami pomocou riešenia postupnosti parametrizovaných úloh na voľný extrém, pričom nájdené minimum aktuálne riešenej podúlohy je použité ako štartovací bod pre ďalšiu minimalizáciu. Existujú aj tzv. primárno-duálne metódy vnútorného bodu, ktoré, na rozdiel od bariérových metód, parametrizujú a riešia Karush-Kuhn-Tuckerove (KKT) podmienky optimality. Hlavnou myšlienkou primárno-duálnych metód vnútorného bodu je, že podmienka komplementarity v KKT podmienkach optimality sa perturbuje kladným parametrom a následne sa rieši postupnosť takto modifikovaných podmienok optimality. Bariérové aj primárno-duálne metódy vnútorného bodu využívajú na riešenie spomínaných parametrizovaných podúloh zvyčajne Newtonovu metódu. Bližšie informácie o histórii a vývoji metód vnútorného bodu možno nájsť napr. v [47], [44], [48], bariérové metódy vnútorného bodu sú spracované napr. v [14],

[80] a primárno-duálne metódy vnútorného napr. v [14], [105], [59].

Metódy pre veľkorozmerné úlohy so špeciálnou štruktúrou

V súčasnej dobe mnohé praktické problémy vedú na veľkorozmerné úlohy voľnej optimalizácie so špeciálnou štruktúrou. Pri riešení takýchto úloh však klasické metódy voľnej optimalizácie nemusia vždy vyhovovať, či už kvôli samotnej štruktúre úlohy, nárokom na čas, alebo pamäť. Preto je prirodzená snaha navrhovať sofistikovanejšie metódy, ktoré by boli schopné lepšie využívať prípadnú špecifickú štruktúru úloh.

Napríklad, účelová funkcia má často tvar súčtu (priemeru) veľkého množstva funkcií merajúcich chybu, ktorú chceme minimalizovať. Medzi tieto úlohy patrí napríklad lineárna, nelineárna a logistická regresia. Ak je počet týchto funkcií veľmi veľký, tak aj výpočet samotného gradientu účelovej funkcie (ktorý je súčtom veľkého množstva gradientov čiastkových funkcií) môže byť výpočtovo náročný. Z toho dôvodu pre minimalizáciu účelovej funkcie v tvare súčtu (priemeru) funkcií vznikli, tzv. prírastkové [10], [11] a stochastické metódy [93], [18], [92], [60], [76], [27], ktoré pozostávajú z výpočtovo jednoduchších iterácií používajúcich jednotlivé časti účelovej funkcie.

Často sa vyskytujú úlohy, v ktorých je účelová funkcia síce nediferencovateľná, ale rozložiteľná na konvexnú diferencovateľnú časť a konvexnú nediferencovateľnú časť. Klasický spôsob ako riešiť nediferencovateľné úlohy na voľný extrém predstavujú subgradientné metódy, ktoré sú vo všeobecnosti pomalé (majú pomalú rýchlosť konvergenzie). Na optimalizáciu úloh na voľný extrém s nediferencovateľnými konvexnými funkciami tejto štruktúry sú vhodné tzv. proximálne metódy. Najznámejšou a najpopulárnejšou proximálnou metódou je tzv. proximal-gradientná metóda [82], [5], [4]. Proximálne metódy možno chápať ako určité zovšeobecnenie klasických metód umožňujúce minimalizáciu konvexných nediferencovateľných funkcií ([82], [66]). Iné proximálne metódy možno zasa vnímať ako modifikácie proximal-gradientnej metódy pre špeciálne typy funkcií ([76], [27]). V súčasnosti prebiehajú snahy o modifikovanie algoritmov rôznych existujúcich metód do proximálneho tvaru, aby sa rozšírila schopnosť metód vysporiadať sa s nehladkými účelovými funkciami.

Poznamenajme, že úlohy na voľný extrém s nehladkou účelovou funkciou je tiež možné v mnohých prípadoch transformovať na hladké úlohy s ohraňovaniami. Na

riešenie týchto úloh zase môžu byť efektívne použité metódy vnútorného bodu.

Úlohy na voľný extrém oboch spomínaných štruktúr, t.j. úlohy s účelovou funkciou v tvare súčtu funkcií a s účelovou funkciou rozložiteľnou na konvexnú diferencovateľnú a konvexnú nediferencovateľnú časť, majú rozsiahle uplatnenie napríklad v štatistike a strojovom učení (pri aproximácii a fitovaní dát, klasifikácii), v oblasti spracovania signálov (zvuk, obrázky, video) a optimálneho dizajnu. Za najvýznamnejších predstaviteľov úloh tohto typu možno považovať tzv. l_1 -regulované úlohy, konkrétne úlohy l_1 -regulovaných najmenších štvorcov a l_1 -regulovanej logistickej regresie, ktorým sa budeme v tejto práci bližšie venovať. Na tieto úlohy budeme aplikovať obidve vyššie uvedené metódy riešenia: jednak rôzne modifikované proximálne metódy, jednak metódy vnútorného bodu a ich modifikácie.

Cieľ a štruktúra práce

Témou dizertačnej práce sú moderné metódy konvexnej optimalizácie pre riešenie veľkorozmerných úloh na voľný extrém, ktorých účelová funkcia je nediferencovateľná a má špeciálnu štruktúru. V práci sa sústredíme na tzv. l_1 -regulované úlohy, pod ktoré patria aj úlohy l_1 -regulovaných najmenších štvorcov a l_1 -regulovanej logistickej regresie, ktoré majú veľké uplatnenie v rôznych oblastiach. Cieľom práce je spracovať prehľad známych metód, ich teoretická analýza, navrhnutie prípadných modifikácií a numerických vylepšení, následné otestovanie a porovnanie funkčnosti týchto metód vykonaním numerických experimentov na umelých dátach a aplikácia vybraných metód na reálne problémy. Práca pozostáva z niekoľkých častí.

Kapitola 1 sa venuje úlohám na voľný extrém so špeciálnou štruktúrou, spomíname viaceré aplikácie týchto úloh a metódy riešenia. V tejto kapitole sa sústredíme na l_1 -regulované úlohy ako úloha l_1 -regulovaných najmenších štvorcov a úloha l_1 -regulovanej logistickej regresie, pričom uvedené úlohy predstavíme, uvádzame prehľad článkov iných autorov o danej problematike, kde sme spracovali prehľad aplikácií a takisto prehľad používaných metód riešenia. V ďalších dvoch kapitolách sa venujeme metódam riešenia.

Kapitola 2 sa zaoberá proximálnymi metódami, kde predstavíme proximálny operátor a jeho vlastnosti, takisto vysvetlíme proximal-gradientnú metódu a jej zrýchlenú verziu, proximal-newtonovu metódu, prírastkovú proximal-gradientnú metódu a tzv.

ADMM (Alternating Direction Method of Multipliers) metódu. V tejto kapitole tiež uvádzame 3 spädoité modifikácie zrýchlenej proximal-gradientnej metódy, z toho jedna je náš vlastný prístup. V tejto kapitole sme taktisto navrhli novú proximal-kvázinewtonovu metódu a na základe rôznych interpretácií proximal-gradientnej metódy sme odvodili analogické interpretácie pre proximal-newtonovu metódu a proximal-kvázinewtonovu metódu. V časti o prírastkových metódach sme pre lepšiu interpretáciu a porovnanie s inými metódami prepísali tzv. MISO metódu do ekvivalentného tvaru, v ktorom vystupuje proximálny operátor.

V Kapitole 3 sa venujeme metódam vnútorného bodu, kde uvádzame primárno-duálnu metódu vnútorného bodu, jej myšlienku, primárno-duálny smer optimalizácie a algoritmus. V tejto kapitole takisto predstavíme novú primárno-duálnu metódu vnútorného bodu s redukciou dimenzie, ktorú sme vytvorili špeciálne na riešenie l_1 -regulovaných úloh. Táto metóda kombinuje proximálne metódy s metódami vnútorného bodu a prináša metódam vnútorného bodu niekoľko potenciálnych výhod.

Kapitola 4 je o implementácii metód riešenia, t.j. o implementácii proximálnych metód a primárno-duálnych metód vnútorného bodu na riešenie l_1 -regulovaných úloh, resp. na riešenie úloh l_1 -regulovaných najmenších štvorcov a l_1 -regulovanej logistickej regresie. V časti o implementácii proximálnych metód odvádzame tvary proximálnych operátorov pre naše úlohy a takisto sa venujeme problematike voľby dĺžky kroku. V časti o implementácii metód vnútorného bodu uvádzame 2 spôsoby transformácie l_1 -regulovaných úloh na konvexnú úlohu s diferencovateľnou účelovou funkciou a lineárnymi ohraničeniami. Pre obe tieto ekvivalentné úlohy odvádzame Newtonove systémy pre hľadanie primárno-duálnych smerov optimalizácie, ktoré následne zredukujeme do tvaru, ktorý zefektívňuje príslušné metódy vnútorného bodu. Takisto opisujeme triky využiteľné pri riešení daných systémov a implementácii iteračných metód.

V Kapitole 5 vysvetlíme metodiku testovania metód a zavedieme značenie metód a ich verzií, ktoré budeme používať vo všetkých experimentoch v práci. Účinnosť použitých metód riešenia a ich rôzne vlastnosti otestujeme v Kapitole 6, kde prezentujeme numerické experimenty na vygenerovaných dátach. Metódy riešenia porovnávame a skúmame vplyv zmeny rozmerov úlohy, hodnoty regularizačného parametra a riedkosti dát na výkon metód. V prípade úlohy l_1 -regulovaných najmenších štvorcov testujeme

takisto vplyv počiatocnej vzdialenosti od riešenia, vplyv počtu nenulových prvkov riešenia či vplyv tzv. dynamického rozsahu riešenia. V Kapitole 7 venovanej reálnym aplikáciám l_1 -regulovaných úloh, testujeme funkčnosť vybraných metód riešenia na reálnych problémoch (a dátach) z rôznych oblastí - napríklad z oblasti optimálneho topologického dizajnu, oblasti spracovania signálov, z medicíny, z oblasti kategorizácie textu či oblasti detekovania škodlivých stránok a spamu. Metódy porovnávame na problémoch: rekonštrukcia riedkeho signálu (SSR - Sparse Signal Reconstruction), dizajn mechanicej konštrukcie (TTD - Truss Topology Design) a l_1 -regulovaná logistická regresia.

Kapitola 1

Úlohy so špeciálnou štruktúrou

V súčasnej dobe mnohé praktické problémy vyžadujú riešiť veľkorozmerné úlohy na voľný extrém špeciálnej štruktúry. Kvôli tejto špeciálnej štruktúre klasické metódy voľnej optimalizácie nie vždy postačujú, a preto je snaha konštruovať a skúmať špeciálne metódy, ktoré vedú čo najlepšie využívať štruktúru úlohy. V tejto práci sa venujeme úlohám na voľný extrém špeciálnej štruktúry, ktorých účelová funkcia sa dá rozložiť na dve časti $f(x)$ a $\lambda g(x)$, t.j.

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + \lambda g(x), \quad (1.1)$$

kde funkcia $f(x)$ je konvexná, diferencovateľná a funkcia $g(x)$ je konvexná a častokrát nediferencovateľná. Parameter $\lambda > 0$ zo zápisu (1.1) predstavuje relatívnu váhu (trade-off parameter), meníme ním váhu kladenú na dva minimalizačné ciele reprezentované funkciami f a g . Takúto úlohu možno teda chápať ako skalarizáciu úlohy bikriteriálnej optimalizácie s dvomi účelovými funkciami f a g .

Úlohy typu (1.1) možno nájsť napríklad pri odhadovaní riedkej inverznej kovariančnej matice (sparse inverse covariance estimation) [39], pri kompletizácii matice pomocou nukleárnej normy (trace-norm matrix completion) [20], [98], či v problémoch s maximovou normou [65]. Medzi ďalšie aplikácie (1.1) patrí aj tzv. elastoplastická úloha vyskytujúca sa pri analýze mechanických štruktúr [52]. Úlohy v tvare (1.1) sú rozšírené predovšetkým v štatistike pri fitovaní dát a v oblasti spracovania signálov (napr. [5]), kde funkcia $f(x)$ často meria nepresnosť medzi nejakými dátami a výstupmi parametrického modelu, ktorého vektor parametrov je $x \in \mathbb{R}^n$.

1.1 Štruktúra úlohy (1.1) z pohľadu f

Na štruktúru úlohy (1.1) sa dá nazerať dvoma spôsobmi - cez funkciu f a cez funkciu g . Pozrime sa najskôr na (1.1) **cez funkciu f** (tento pohľad zahŕňa aj prípad $g(x) \equiv 0$). Mnohé úlohy na voľný extrém vyskytujúce sa v štatistike a v strojovom učení majú funkciu f tvaru

$$f(x) = \frac{1}{m} \sum_{i=1}^m f_i(x),$$

t.j. hľadáme bod minima konečného súčtu hladkých konvexných funkcií (resp. priemeru m funkcií), ktoré merajú nepresnosť vo veľa dátových bodoch (meraniach / pokusoch), pričom m označuje počet pokusov. Klasický príklad úlohy tohto typu je tzv. úloha najmenších štvorcov, ktorá súvisí s úlohou lineárnej regresie. Pri úlohe najmenších štvorcov je funkcia f tvaru

$$f(x) = \frac{1}{m} \sum_{i=1}^m (a_i^T x - b_i)^2,$$

kde $a_i \in \mathbb{R}^n$ a $b_i \in \mathbb{R}$ sú dátové vzorky. Ďalší dôležitý príklad je tzv. úloha logistickej regresie, pri ktorej má funkcia f tvar

$$f(x) = \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i a_i^T x)),$$

kde $a_i \in \mathbb{R}^n$ a $b_i \in \{-1, 1\}$ sú dátové vzorky prislúchajúce binárnej klasifikácii.

V súčasnosti existujú aplikácie, kde je obrovské množstvo dátových vzoriek, t.j. m je veľmi veľké. V gradientnej metóde je v každom kroku potrebné vyčíslovať gradient funkcie f , ktorý pozostáva zo súčtu m gradientov čiastkových funkcií. Z toho dôvodu boli vyvinuté špeciálne metódy schopné lepšie využívať túto štruktúru úlohy, ktoré pozostávajú z výpočtovo jednoduchších iterácií používajúcich jednotlivé čiastkové funkcie f_i . Medzi tieto metódy patria tzv. prírastkové metódy (angl. incremental methods) [10], [11]. Tento špeciálny prípad úlohy na voľný extrém je takisto vhodný aj na použitie metód využívajúcich náhodnosť, tzv. stochastických metód. Populárnou a najjednoduchšou metódou využívajúcou náhodnosť je stochastická gradientná metóda (angl. stochastic gradient descent)[77], [93], [18], ktorá ako nevychýlený odhad gradientu funkcie $f(x)$ používa gradient náhodne vybranej čiastkovej funkcie $f_i(x)$. Medzi vyvinutejšie a modernejšie stochastické metódy patria napríklad semi-stochastická gradientná

metóda [60], SAG (stochastic average gradient) metóda [92], MISO (Minimization by Incremental Surrogate Optimization) metóda [75], [76], či SAGA metóda [27].

1.2 Štruktúra úlohy (1.1) z pohľadu g

Teraz sa pozrime na úlohu (1.1) **cez funkciu g** . Funkcia $g(x)$ z úlohy (1.1) častokrát predstavuje penalizáciu alebo regularizáciu, ktorá indukuje požadované vlastnosti riešenia. Rôzne typy regularizácii používaných pri spracovaní signálov možno nájsť napr. v [5]. Medzi úlohy tohto typu patrí aj úloha regulovaných najmenších štvorcov či úloha regulovanej logistickej regresie.

Štandardným typom regularizácie je tzv. Tikhonovova regularizácia (tiež zvaná l_2 regularizácia) $g(x) = \|x\|_2^2$ štandardne používaná proti tzv. over-fitu. Over-fit nastáva vtedy, keď má štatistický model príliš veľa parametrov a tak okrem skúmaného javu model popisuje aj šum prítomný v meraniach. Takéto modely majú častokrát následne slabú predikčnú schopnosť a pri použití modelu na dodatočné dáta model už dobre nefunguje. Tikhonovova, t.j. kvadratická regularizácia je bezproblémová - umožňuje použitie metód pre diferencovateľné funkcie.

V súčasnosti sa stali veľmi populárne nediferencovateľné regularizačné funkcie. Najvýznamnejším predstaviteľom nediferencovateľných regularizácií je tzv. l_1 regularizácia, kde $g(x) = \|x\|_1$. Táto l_1 regularizácia sa využíva tiež proti over-fitu ale aj ako penalizácia veľkých hodnôt riešenia. Veľkou výhodou l_1 regularizácie je, že indukuje riedkosť vektora riešenia x (t.j. x má malý počet nenulových zložiek) [97]. Vďaka l_1 regularizácii sa podarilo odstrániť problém Tikhonovovej regularizácie, pri ktorej sú zvyčajne všetky zložky riešenia x nenulové. Z toho dôvodu sa l_1 norma (l_1 regularizácia) používa ako aproximácia l_0 „normy“¹ (l_0 regularizácie). Úlohy využívajúce l_0 normu sú teoreticky atraktívne, no vo všeobecnosti sú NP ťažké a vyžadujú kombinatorickú optimalizáciu. Úlohy s l_1 normou sú lepšie riešiteľné, l_1 regularizácia preto predstavuje užitočný nástroj pre riešenie úloh, pri ktorých požadujeme riedky vektor riešenia.

Ako sme už spomínali, funkcia g z úlohy (1.1) je konvexná a častokrát nediferenco-

¹ l_0 norma je rovná počtu nenulových zložiek vektora, t.j. $\|x\|_0 = \#\{i | x_i \neq 0\}$. l_0 norma nespĺňa vlastnosti normy, a preto nie je norma. Napriek tomu sa tak v literatúre označuje a nazýva.

vateľná. Nediferencovateľnosť účelovej funkcie predstavuje hlavný problém pri riešení úloh takejto štruktúry. Klasický spôsob ako riešiť nediferencovateľné úlohy na voľný extrém predstavujú subgradientné metódy. Tieto metódy sú však vo všeobecnosti veľmi pomalé, sú citlivé na voľbu dĺžky kroku a nie sú spádové. Na optimalizáciu úloh na voľný extrém s nediferencovateľnými konvexnými funkciami tejto štruktúry sú vhodné tzv. proximálne metódy. Základným a spoločným prvkom algoritmov proximálnych metód je tzv. proximálny operátor (angl. proximal operator). Mnohé proximálne metódy môžeme vnímať ako určité zovšeobecnenie klasických metód umožňujúce riešiť úlohy s nediferencovateľnými funkciami - napr. proximal-gradientnú metódu [82], [5], [4] ako zovšeobecnenie gradientnej metódy alebo proximal-newtonovu metódu [66] ako zovšeobecnenie Newtonovej metódy. Iné proximálne metódy možno vnímať ako modifikácie proximal-gradientnej metódy pre špeciálne typy funkcií - napr. už spomínaná MISO metóda [76], [75] a SAGA metóda [27] sú prispôsobené pre $F(x)$ v tvare priemeru funkcií. Medzi proximálne metódy patrí taktiež ADMM metóda [16], [82]. V súčasnej dobe prebiehajú snahy o modifikovanie algoritmov rôznych existujúcich metód do tzv. proximálneho tvaru, aby sa rozšírila schopnosť metód vysporiadať sa s nehladkými účelovými funkciami. Úlohy na voľný extrém s nehladkou účelovou funkciou je taktiež možné v mnohých prípadoch (napr. pri spomínanej l_1 regularizácii) transformovať na hladké úlohy s ohraničeniami. Na riešenie týchto úloh môžu byť následne efektívne použité metódy vnútorného bodu [48], [14], [105], [80], [59].

V ďalšej časti práce sa sústreďíme na l_1 -regulované úlohy (1.1), t.j. také kde $g(x) = \|x\|_1$. Konkrétne sa budeme venovať úlohe l_1 -regulovaných najmenších štvorcov a úlohe l_1 -regulovanej logistickej regresie. V týchto úlohách je $\lambda > 0$ kladný regularizačný parameter, ktorý predstavuje trade-off parameter medzi minimalizovaním $f(x)$ a $g(x) = \|x\|_1$.

1.3 Úloha l_1 -regulovaných najmenších štvorcov

Mnohé problémy z oblasti štatistiky, spracovania signálu, či z iných oblastí sa dajú opísať nasledujúcim lineárnym modelom tvaru

$$b = Ax + e,$$

kde $b \in \mathbb{R}^m$ je vektor pozorovaní, $A \in \mathbb{R}^{m \times n}$ je dátová matica, $x \in \mathbb{R}^n$ je vektor neznámych parametrov modelu, ktorý hľadáme, a vektor $e \in \mathbb{R}^m$ predstavuje šum v pozorovaní. Riadky a_i matice A často predstavujú namerané hodnoty vstupných premenných a zložky b_i vektora b predstavujú k nim prislúchajúce hodnoty výstupných premenných.

V prípade $m < n$ (počet pozorovaní je menší ako počet neznámych), klasická regresia pomocou riešenia úlohy najmenších štvorcov vedie k tzv. over-fitu - výsledný vektor x popisuje aj šum v pozorovaniach, čo je nežiadúci jav. Štandardnou štatistickou technikou proti over-fitting-u je pridanie regularizačného člena do účelovej funkcie úlohy najmenších štvorcov. Namiesto štandardnej Tikhonovovej (l_2) regularizácie sa venujeme tzv. l_1 regularizácii, čím dostaneme úlohu l_1 -regulovaných najmenších štvorcov

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1, \quad (1.2)$$

kde $\lambda > 0$ je kladný regularizačný parameter.

1.3.1 Vlastnosti úlohy l_1 -regulovaných najmenších štvorcov

Optimálne riešenie úlohy (1.2) vždy existuje, no nemusí byť jednoznačné [56]. Na rozdiel od Tikhonovovej regularizácie, úloha (1.2) nemá analytické riešenie. Riešenie musí byť počítané numericky.

Medzi najdôležitejšie vlastnosti úlohy (1.2) patrí fakt, že vďaka prítomnosti l_1 normy v účelovej funkcii úloha produkuje riedky vektor riešenia x , t.j. x obsahuje málo nenulových zložiek. Regularizačný parameter λ kontroluje trade-off medzi riedkosťou riešenia a veľkosťou chyby (šumu) riešenia. Zvyčajne platí, že so zväčšujúcou sa hodnotou regularizačného parametra λ vo vektore riešenia x , pribúda počet nulových zložiek (zväčšuje sa riedkosť riešenia) [97]. Pre hodnoty regularizačného parametra

$$\lambda \geq \lambda_{max} = \|A^T b\|_\infty \quad (1.3)$$

je už optimálnym riešením úlohy (1.2) nulový vektor ($x^* = 0$). Hraničná hodnota λ_{max} vyplýva priamo z podmienok optimality úlohy (1.2) [56].

Riedky vektor riešenia úlohy (1.2) možno taktiež považovať za výber dôležitých vstupných premenných modelu (tzv. feature selection). Nenulové zložky riešenia (parametre) prislúchajú signifikantným vstupným premenným a nulové prislúchajú menej podstatným premenným. Úloha l_1 -regulovaných najmenších štvorcov týmto spôsobom vyberie vhodný model. Navyše, riedke riešenia častokrát v praxi poskytujú lepšiu interpretáciu.

Úloha l_1 -regulovaných najmenších štvorcov (1.2) je ekvivalentná s úlohou

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|Ax - b\|_2 \\ \text{s.t.} \quad & \|x\|_1 \leq \epsilon_1, \end{aligned}$$

a tiež s úlohou

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|x\|_1 \\ \text{s.t.} \quad & \|Ax - b\|_2 \leq \epsilon_2, \end{aligned}$$

pre špecifické hodnoty parametrov λ , ϵ_1 , ϵ_2 [38]. Parametre ϵ_1 , ϵ_2 regulujú riedkosť riešenia, resp. ohraničujú veľkosť chyby (šumu) riešenia. Prvá z týchto úloh je tiež známa pod názvom LASSO (Least Absolute Shrinkage and Selection Operator), ktorú v roku 1996 predstavil R. Tibshirani v [97]. Z toho dôvodu sa úlohe l_1 -regulovaných najmenších štvorcov hovorí v štatistike aj LASSO regresia.

Ďalšie vlastnosti úlohy (1.2) možno nájsť napr. v [56], [63].

1.3.2 Prehľad aplikácií a metód riešenia úlohy l_1 -regulovaných najmenších štvorcov

Úloha l_1 -regulovaných najmenších štvorcov (1.2) má v praxi široké uplatnenie, v jej tvare sa dá naformulovať mnoho problémov. Jednou z prvých oblastí, kde sa začala využívať l_1 regularizácia a úloha l_1 -regulovaných najmenších štvorcov bola štatistika. Napríklad, v [97] autor predstavil LASSO úlohu, ktorú navrhol ako nástroj pre výber vstupných premenných modelu za účelom získania lepšie interpretovateľných modelov. Úloha l_1 -regulovaných najmenších štvorcov sa používa na fitovanie a kompresiu dát, resp. na nájdenie riedkych reprezentácií, v strojovom učení [12], [28]. V oblasti

spracovania signálu je úloha (1.2) známa taktiež ako Basis Pursuit Denoising, ktorú predstavili v [25]. Úloha (1.2) sa používa na rekonštrukciu signálu napríklad v [22], [28], [31], [106], [5], takisto sa dá použiť pri zobrazovaní magnetickej rezonancie [56], [73] či pri zaostrovaní obrázkov [49], [4]. Úloha tvaru (1.2) bola využitá taktiež pri študovaní DNA s cieľom preskúmať faktory, na ktorých závisí ochorenie [49]. V tvare úlohy (1.2) sa dá naformulovať aj metóda na odhadovanie podkladového trendu v časových radoch nazývaná l_1 trend filtering [55], [63]. Medzi ďalšie aplikácie úlohy (1.2) patrí aj dizajn mechanických konštrukcií (angl. Truss Topology Design) [85].

Na riešenie úlohy l_1 -regulovaných najmenších štvorcov (1.2), bola v [56] použitá primárna bariérová metóda vnútorného bodu, kde sa na výpočet Newtonoveho smeru používajú prepodmienené iteračné metódy. Metódy vnútorného bodu s využitím prepodmienených iteračných metód na riešenie (1.2) taktiež možno nájsť v [38], [89]. Metódy vnútorného bodu takisto využíva solver l_1 -magic [21], kde je namiesto (1.2) riešená druhá z ekvivalentných formulácií uvedených v predošlej časti. Medzi metódy využívajúce informáciu druhého rádu taktiež patria pdNCG (primal-dual Newton Conjugate Gradients) metóda [37], Projected Scaled Subgradient metóda [90] a takisto Dual Projected Newton metóda [109], kde je však potrebná regularita matice $A^T A$.

Medzi metódy prvého rádu na riešenie (1.2) patrí napríklad proximálna metóda FISTA (Fast Iterative Shrinkage Thresholding Algorithm) [4], paralelné súradnicovo spádové metódy (angl. parallel coordinate descent methods) [86], NestA [6], či proximálna metóda SpaRSA (Sparse Reconstruction by Separable Approximation) [106], ktorá využíva na určenie dĺžky kroku praktiky dvojkrokových metód Barzilai-Borwein [3]. Barzilai-Borwein metódami je inšpirovaná aj GPSR (Gradient Projection for Sparse Reconstruction) metóda [35] a takisto metóda v [17]. Úloha v tvare (1.2) bola úspešne riešená paralelnými súradnicovo spádovými metódami v [85], kde boli riešené úlohy obrovských rozmerov využitím výpočtového výkonu grafických kariet. Porovnanie vybraných metód prvého a druhého rádu (FISTA, parallel coordinate descent m., Projected Scaled Subgradient m., primal-dual Newton Conjugate Gradients m.) možno nájsť v [36].

Poslednou triedou metód, ktoré sa používajú na riešenie úlohy (1.2) sú tzv. homotopické metódy, ktoré využívajú fakt, že úloha (1.2) má tzv. po častiach lineárnu

regularizačnú cestu pre parameter λ [56]. Tieto metódy spravidla sledujú túto cestu (začína sa v λ_{max} a bode $x = 0$) až po požadovanú hodnotu parametra λ . Medzi homotopické metódy patrí napríklad tzv. LARS (Least-Angle Regression) metóda [32], proximal-gradientná homotopická metóda [107], či metódy v [30],[50].

1.4 Úloha l_1 -regulovanej logistickej regresie

V tejto podkapitole sa budeme najskôr venovať úlohe logistickej regresie. Potom sformulujeme a opíšeme vlastnosti samotnej l_1 -regulovanej logistickej regresie a následne uvedieme prehľad aplikácií a metód riešenia.

1.4.1 Úloha logistickej regresie

Logistická regresia hľadá nadrovinu, ktorá najlepšie separuje dáta patriace do dvoch tried (jedná sa preto o tzv. lineárnu klasifikáciu). Predpokladajme, že sme dostali tzv. trénovaciu vzorku dát, t.j. namerané dátové body

$$(a_i, b_i) \in \mathbb{R}^n \times \{-1, +1\}, \quad i = 1, \dots, m,$$

kde a_i sú vektory charakteristík (vstupné/vysvetľujúce premenné, angl. feature variables), a premenná b_i označuje prislúchajúcu triedu (binárny výstup). Predpokladáme, že tieto vzorky sú nezávislé rovnako rozdelené.

Chceme nájsť separujúcu nadrovinu $\{x^T a + y = 0\}$ takú, že $x^T a_i + y > 0$ ak $b_i = 1$ a $x^T a_i + y < 0$ ak $b_i = -1$. Hľadáme teda jej parametre x, y , pričom $x \in \mathbb{R}^n$ nazývame vektorom váh a $y \in \mathbb{R}$ je priesečník. Logistický model má nasledujúci tvar

$$p(b|a) = \frac{\exp(b(x^T a + y))}{1 + \exp(b(x^T a + y))} = \frac{1}{1 + \exp(-b(x^T a + y))}, \quad (1.4)$$

kde $p(b|a)$ je podmienená pravdepodobnosť triedy b pri daných charakteristikách $a \in \mathbb{R}^n$. Označme podmienené pravdepodobnosti pre naše dáta ako $p_i(x, y) = p(b_i|a_i)$, $i = 1, \dots, m$, kde x, y sú parametre logistického modelu. Hodnoty parametrov x, y separujúcej nadroviny následne nájdeme (odhadneme) pomocou metódy maximálnej virohodnosti, kde je potrebné maximalizovať funkciu virohodnosti (angl. likelihood function) $\prod_{i=1}^m p_i(x, y)$, resp. logaritmus funkcie virohodnosti (angl. log-likelihood function) $\sum_{i=1}^m \log p_i(x, y)$ (ide o monotónnu transformáciu).

Úloha logistickej regresie preto vyzerá

$$\min_{x,y} -\frac{1}{m} \sum_{i=1}^m \log p_i(x, y) = \min_{x,y} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T x + y))), \quad (1.5)$$

kde $x \in \mathbb{R}^n$ a $y \in \mathbb{R}$ sú premenné. Účelová funkcia úlohy logistickej regresie (nazývaná average logistic loss) je diferencovateľná a konvexná. Z trénovacej vzorky teda vieme pomocou riešenia úlohy na voľný extrém nájsť parametre $\hat{x}, \hat{y}$ separujúcej nadroviny, tzv. odhady metódou maximálnej vierohodnosti.

Poznámka 1.4.1. Pre lepšie pochopenie logistickej regresie je dobré uvedomiť si, čo sa deje pri minimalizácii účelovej funkcie úlohy (1.5). Označme funkciu $f = \log(1 + \exp(-b(a^T x + y)))$ a rozdelme úvahu na dva prípady:

- Prípad triedy $b = 1$

$$\text{Ak } p(1|a) = \frac{1}{1 + \exp(-(x^T a + y))} \rightarrow 1, \text{ tak } f \rightarrow 0.$$

$$\text{Ak } p(1|a) = \frac{1}{1 + \exp(-(x^T a + y))} \rightarrow 0, \text{ tak } f \rightarrow \infty.$$

- Prípad triedy $b = -1$

$$\text{Ak } p(-1|a) = \frac{1}{1 + \exp(x^T a + y)} \rightarrow 1, \text{ tak } f \rightarrow 0.$$

$$\text{Ak } p(-1|a) = \frac{1}{1 + \exp(x^T a + y)} \rightarrow 0, \text{ tak } f \rightarrow \infty.$$

Z predchádzajúceho vidíme, že prípad, keď je trieda $b = 1$ a predpovedáme $p(1|a) \approx 0$, penalizujeme veľmi veľkými hodnotami funkcie f . Analogicky penalizujeme v prípade $b = -1$ a predpovede $p(-1|a) \approx 0$. Z toho dôvodu môžeme logistickú regresiu (1.5) interpretovať ako úlohu, kde hľadáme parametre x, y tak, aby sme minimalizovali chybné zaradenie do tried na danej dátovej vzorke.

Poznámka 1.4.2. Účelová funkcia v (1.5) je vždy nezáporná. Jej optimálna hodnota leží v intervale $[0, 1]$, pričom nulovú hodnotu dosahuje len v prípade, ak sú dané dáta lineárne separovateľné [58].

Ak nájdeme riešenie úlohy (1.5), t.j. nájdeme hodnoty parametrov $\hat{x}, \hat{y}$ separujúcej nadroviny $\{\hat{x}^T a + \hat{y} = 0\}$, vieme pre nový vektor charakteristík $a \in \mathbb{R}^n$ predpovedať pravdepodobnosti zaradenia do triedy $b = 1$ ($b = -1$) na základe logistického modelu

(1.4). Takisto môžeme novému dátovému bodu a prideliť triedu pomocou lineárneho klasifikátora (ak $\hat{x} \neq 0$)

$$K(a) = \begin{cases} +1 & \text{ak } \hat{x}^T a + \hat{y} > 0 \\ -1 & \text{ak } \hat{x}^T a + \hat{y} \leq 0 \end{cases}, \quad (1.6)$$

ktorý pre daný vektor a vyberie viac pravdepodobnú triedu na základe logistického modelu.

1.4.2 Formulácia a vlastnosti úlohy l_1 -regulovanej logistickej regresie

Podobne ako v prípade úlohy najmenších štvorcov, aj v prípade úlohy logistickej regresie sa častokrát využíva regularizácia, aby sa vyhlo overfitting-u a aby sa získal robustnejší klasifikátor [94].

Úloha l_1 -regulovanej logistickej regresie vyzerať nasledovne

$$\min_{x,y} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T x + y))) + \lambda \|x\|_1, \quad (1.7)$$

kde $x \in \mathbb{R}^n$ a $y \in \mathbb{R}$ sú premenné a $\lambda > 0$ je kladný regularizačný parameter.

Poznámka 1.4.3. Všimnime si, že l_1 penalizácia je aplikovaná len na vektor váh x a nie na priesečník y , pretože tento parameter má dôležitú úlohu v lineárnom klasifikátore (1.6) (pri určovaní tried).

Optimálne riešenie úlohy (1.7) vždy existuje, no nemusí byť jednoznačné [58]. Účelová funkcia úlohy (1.7) je konvexná, avšak nie je diferencovateľná, čo predstavuje hlavný problém pri riešení tejto úlohy. Napriek tomu je o úlohu l_1 -regulovanej logistickej regresie veľký záujem, predovšetkým kvôli nasledujúcej atraktívnej vlastnosti. Vďaka prítomnosti l_1 normy v účelovej funkcii, úloha (1.7) zvyčajne poskytuje riedky vektor váh x (t.j. x obsahuje málo nenulových zložiek). Tým sa líši od štandardnej l_2 regularizácie, kde sú zvyčajne všetky zložky riešenia x nenulové. Regularizačný parameter λ riadi počet nenulových zložiek vektora váh. Väčšinou platí, že pri zväčšujúcej sa hodnote regularizačného parametra λ pribúdajú nulové zložky vo vektore váh x (vektor váh sa stáva redším) [97], [58].

Podobne ako pri úlohe (1.2) aj pri úlohe (1.7) existuje hraničná hodnota λ_{max} . Podľa [58], hraničná hodnota

$$\lambda_{max} = \frac{1}{m} \|A^T \tilde{b}\|_{\infty}, \quad (1.8)$$

kde

$$\tilde{b}_i = \begin{cases} m_-/m & b_i = 1 \\ -m_+/m & b_i = -1, \end{cases} \quad i = 1, \dots, m, \quad (1.9)$$

kde m_- (m_+) je počet vzoriek s výsledkom -1 (1). Pre hodnoty regularizačného parametra $\lambda \geq \lambda_{max}$ je už optimálnym riešením úlohy (1.7) nulový vektor váh $x^* = 0$.

Riedke riešenia úlohy (1.7) sa dajú taktiež chápať ako výber dôležitých vstupných premenných (charakteristík) modelu. Ak je j -ta zložka vektora váh x nulová, znamená to, že prislúchajúci logistický model nevyužíva j -tu zložku vektora charakteristík a . Preto k riedkemu vektoru váh prislúcha logistický model, ktorý využíva iba niektoré charakteristiky, čo možno považovať za výber dôležitých vstupných premenných modelu. Častokrát požadujeme, aby separujúcu nadrovinu a následnú klasifikáciu ovplyvňovali len signifikantné premenné. Navyše, riedke riešenia a k nim zodpovedajúce modely majú častokrát v praxi lepšiu interpretáciu. Táto vlastnosť je dôležitá napríklad pri analýze medicínskych dát.

1.4.3 Prehľad aplikácií a metód riešenia úlohy l_1 -regulovanej logistickej regresie

Logistická regresia predstavuje populárny nástroj pre binárnu klasifikáciu v strojovom učení [12]. Úloha logistickej regresie je používaná v oblasti bioinformatiky [100] a nachádza rozsiahle uplatnenie pri analýze medicínskych dát [51], [57], [68]. Napríklad, v [70] sa autori klasifikáciou pomocou génov snažia určiť typ rakoviny, v [83] bola úloha logistickej regresie použitá na analýzu EEG a v publikácii [95] je logistická regresia použitá na predpovedanie úmrtnosti do 30 dní po akútnom infarkte myokardu. Úloha l_1 -regulovanej logistickej regresie (1.7) bola skúmaná aj v [97], kde boli ukázané dobré vlastnosti l_1 -regularizácie ako robustnosť vzhľadom na šum a schopnosť vyberať dôležité vstupné premenné pre model.

Na riešenie úlohy l_1 -regulovanej logistickej regresie (1.7), bola v [58] použitá bariérová metóda vnútorného bodu, kde sa na výpočet Newtonovho smeru používa prepodmienená metóda konjugovaných gradientov. Tzv. IRLS-LARS (iteratively reweighted least squares least angle regression) metóda z [67] používa v každej iterácii kvadratickú aproximáciu účelovej funkcie úlohy (1.7), ktorá je následne riešená už spomínanou LARS metódou [32] pre úlohu (1.2). Autori v článku [94] na riešenie úlohy (1.7) navrhli hybridný algoritmus, kde sa v prvej fáze používa metóda prvého rádu (modifikovaná proximal-gradientná metóda) a v druhej fáze sa používa metóda vnútorného bodu navrhnutá v [58]. Hybridný algoritmus sa snaží zachovať najlepšie vlastnosti z oboch svetov - výpočtovú nenáročnosť metód prvého rádu v počiatočných iteráciách a rýchlosť konverencie metód druhého rádu v záverečných iteráciách. Medzi moderné metódy využívajúce náhodnosť na riešenie úlohy (1.7) patrí napríklad už spomínaná proximálna metóda MISO [76], [75]. Zástupcom proximálnych metód a metód druhého rádu na riešenie úlohy (1.7) je proximal-newtonova metóda z [66]. V [16] je distributívna verzia ADMM metódy, ktorá umožňuje riešiť veľkorozmerné úlohy tvaru (1.7) rozdelením optimalizácie na viaceré procesory. Paralelizáciu využíva aj súradnicovo spádová metóda v [99], ktorá navyše uplatňuje tzv. Armijo pravidlo na určovanie dĺžky kroku.

V [40] je na úlohu (1.7) aplikovaná tzv. BBR (Bayesian Binary Regression) metóda, ktorá predstavuje implementáciu cyklických súradnicovo spádových metód, a v [87] navrhli zovšeobecnenú LASSO metódu. Ďalšie metódy využiteľné na riešenie úlohy (1.7) možno nájsť v [108], kde sa nachádza taktiež porovnanie viacerých metód na optimalizáciu úloh l_1 -regularizovaných lineárnych klasifikácii (vrátane úlohy (1.7)), porovnanie zahŕňa napríklad bariérovú metódu vnútorného bodu z [58], BBR metódu, rôzne modifikácie súradnicovo spádových metód. Podobný prehľad a porovnanie metód pre riešenie úlohy (1.7) sa nachádza v [91], kde je však porovnaný iba počet iterácii potrebný na dosiahnutie riešenia (v praxi je pre nás dôležitý výpočtový čas).

Časť I

Metódy riešenia

Ako sme už spomenuli v predošlej kapitole, našim cieľom je minimalizovať úlohu na voľný extrém špeciálnej štruktúry

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + \lambda g(x),$$

kde $\lambda > 0$ a účelová funkcia $F(x)$ sa dá rozložiť na dve časti $f(x)$ a $g(x)$, pričom funkcia $f(x)$ je konvexná diferencovateľná a funkcia $g(x)$ je konvexná nediferencovateľná. Konkrétnejšie, budú nás zaujímať dve úlohy, ktoré majú tento tvar: úloha l_1 -regulovaných najmenších štvorcov

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1$$

a úloha l_1 -regulovanej logistickej regresie

$$\min_{x,y} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T x + y))) + \lambda \|x\|_1.$$

Hlavným problémom riešiteľnosti týchto úloh na voľný extrém je nediferencovateľnosť účelovej funkcie spôsobená prítomnosťou funkcie $g(x) = \|x\|_1$. Na základe toho, ako sa vysporiadame s nediferencovateľnosťou, môžeme deliť postupy na priame a nepriame. Prvý postup spočíva v priamom aplikovaní metód vhodných na optimalizáciu nediferencovateľných funkcií na pôvodnú úlohu. Medzi takéto metódy patria napríklad subgradientné metódy, avšak na optimalizáciu funkcií rozložiteľných na konvexnú diferencovateľnú a konvexnú nediferencovateľnú časť sú omnoho vhodnejšie tzv. proximálne metódy. Nepriamy postup pozostáva z transformácie nediferencovateľnej úlohy na konvexný problém s diferencovateľnou účelovou funkciou a lineárnymi ohraňeniami. Na riešenie takýchto úloh s ohraňeniami sú efektívne tzv. metódy vnútorného bodu. V nasledujúcom texte sa preto budeme venovať proximálnym metódam a metódam vnútorného bodu, takisto poukážeme na to, akým spôsobom aplikovať uvedené metódy priamo na naše problémy a uvedieme modifikácie a triky ako možno zlepšiť fungovanie metód v praxi.

Kapitola 2

Proximálne metódy

Štandardný spôsob ako riešiť nediferencovateľné úlohy na voľný extrém predstavujú subgradientné metódy. Tieto metódy sú však veľmi pomalé, sú citlivé na voľbu dĺžky kroku a nie sú spádové. Preto boli na optimalizáciu úloh na voľný extrém s nediferencovateľnými konvexnými funkciami špeciálnej štruktúry vytvorené sofistikovanejšie metódy tzv. proximálne metódy [101], [82]. V prípade väčšiny proximálnych metód sa jedná o metódy prvého rádu, a preto sú častokrát vhodné na riešenie veľkorozmerných úloh. Mnohé z nich sú akoby "proximálnymi" verziami metód, ktoré sú určené pre optimalizáciu diferencovateľných funkcií. Napríklad, proximal-gradientnú metódu môžeme chápať ako modifikáciu gradientnej metódy. V súčasnosti stále prebiehajú snahy o modifikovanie algoritmov rôznych existujúcich metód do proximálneho tvaru, aby sa rozšírila schopnosť metód vysporiadať sa s nehladkými účelovými funkciami.

2.1 Proximálny operátor

Väčšina proximálnych metód je stavaná na riešenie úloh na voľný extrém špeciálnej štruktúry, kde účelová funkcia má konvexnú diferencovateľnú a konvexnú nediferencovateľnú časť. Základným a spoločným prvkom algoritmov proximálnych metód je tzv. proximálny operátor (angl. proximal operator) [101], [82], [62], [63].

Definícia 2.1.1. (Proximálny operátor) [82] Nech funkcia $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ je uzavretá vlastná konvexná funkcia (angl. closed proper convex function)¹. Proximálny

¹Pozri Príloha A.1 - Definícia A.1.1.

operátor $prox_g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ prislúchajúci k funkcii g je definovaný ako:

$$prox_g(x) = \operatorname{argmin}_u \left(g(u) + \frac{1}{2} \|u - x\|_2^2 \right). \quad (2.1)$$

Vďaka tomu, že funkcia $g(u) + \frac{1}{2} \|u - x\|_2^2$ je silnokonvexná², takto definovaný proximálny operátor pre funkciu g existuje a je jednoznačne určený pre každé $x \in \mathbb{R}^n$. Samotné nájdenie hodnoty proximálneho operátora však zahŕňa riešenie konvexnej úlohy, ktorá sa dá riešiť štandardnými metódami. Avšak častokrát sa proximálny operátor dá explicitne vyjadriť alebo rýchlo riešiť jednoduchými špecializovanými metódami, čo je dôležité pre efektívne využitie proximálnych metód.

V algoritmoch proximálnych metód sa používa proximálny operátor funkcie g s parametrom $t > 0$ tvaru

$$prox_{tg}(x) = \operatorname{argmin}_u \left(g(u) + \frac{1}{2t} \|u - x\|_2^2 \right). \quad (2.2)$$

Zo zápisu proximálneho operátora (2.2) vidíme, že ho možno interpretovať ako kompromis medzi blízkosťou k bodu x a minimalizovaním funkcie g , pričom parameter t možno chápať ako relatívnu váhu medzi týmito cieľmi. Čiže bod $x^+ = prox_{tg}(x)$ je bližšie k minimu funkcie g , pričom parameter t určuje dĺžku tohto kroku. Takisto si všimnime, že keby bol bod x mimo definičného oboru funkcie g , následnou aplikáciou proximálneho operátora už dostaneme bod, ktorý tam patrí.

V Tabuľke 2.1. uvádzame tvary proximálnych operátorov pre vybrané funkcie. Tieto a ďalšie tvary proximálnych operátorov možno nájsť napr. v [82], odvodenie uvedených tvarov proximálnych operátorov sa nachádza v autorovej bakalárskej práci [62].

Z Tabuľky 2.1. si všimnime, že pre indikátorovú funkciu $I_C(u)$ je proximálny operátor Euklidovskou projekciou na množinu C . Proximálne operátory majú blízke prepojenie s projekciami a možno ich považovať za zovšeobecnené projekcie [82].

Jednou z najdôležitejších vlastností proximálnych operátorov je nasledujúca vlastnosť, ktorá umožňuje jednoduchšie odvodenie explicitných tvarov proximálnych operátorov a je takisto kľúčom k paralelizácii proximálnych algoritmov.

Tvrdenie 2.1.1. [82], [62] *Ak g je plne separovateľná funkcia, t.j. $g(x) = \sum_{i=1}^n g_i(x_i)$, potom pre proximálny operátor platí*

$$(prox_g(x))_i = prox_{g_i}(x_i).$$

²Pozri Príloha A.1 - Definícia A.1.3.

Funkcia $g(u)$:	Predpis:	Proximálny operátor $prox_{tg}(x)$:
Kvadratická funkcia	$\frac{1}{2}u^T Au + b^T u + c$	$(tA + I)^{-1}(x - tb)$
l_1 norma	$\ u\ _1 = \sum_{i=1}^n u_i $	$(x - t)_+ - (-x - t)_+$
Euklidovská norma	$\ u\ _2$	$x(1 - \frac{t}{\ x\ _2})_+$
Elastic net	$\ u\ _1 + \frac{\gamma}{2}\ u\ _2^2$	$\frac{1}{1+t\gamma}prox_{t\ \cdot\ _1}(x)$
Logaritmická bariéra	$-\sum_{i=1}^n \log u_i$	$prox_{tg}(x)_i = \frac{x_i + \sqrt{x_i^2 + 4t}}{2}$
Indikátorová funkcia	$I_C(u)$	$P_C(x) = \operatorname{argmin}_{u \in C} \ u - x\ _2^2$

Tabuľka 2.1: Tvary proximálnych operátorov pre niektoré funkcie.

Ďalšia dôležitá vlastnosť proximálnych operátorov je uvedená v nasledujúcej vete a poukazuje na prepojenie proximálneho operátora a subgradientu³ funkcie g . Táto vlastnosť bude hrať dôležitú úlohu pri interpretácii proximálnych metód, a preto ju uvádzame aj s dôkazom.

Veta 2.1.1. [82], [101] *Proximálny operátor $prox_{tg}$ a subdiferenciálny operátor ∂g sú prepojené vzťahom*

$$prox_{tg} = (I + t\partial g)^{-1}. \quad (2.3)$$

Dôkaz. Predpokladajme, že funkcia g je subdiferencovateľná na svojom definičnom obore. Stačí nám ukázať, že $y = prox_{tg}(x)$ práve vtedy, keď $y \in (I + t\partial g)^{-1}(x)$. Proximálny operátor existuje a je jednoznačný pre každé $x \in \mathbb{R}^n$, využitím definície máme

$$y = prox_{tg}(x) = \operatorname{argmin}_u \left(g(u) + \frac{1}{2t} \|u - x\|_2^2 \right).$$

Následne z nutných a postačujúcich podmienok optimality pre subdiferencovateľné funkcie⁴ vyplýva

$$0 \in \partial g(y) + \frac{1}{t}(y - x),$$

z čoho prenásobením parametrom t a preusporiadaním získame vzťah

$$x \in y + t\partial g(y) = (I + t\partial g)(y).$$

Preto platí $y \in (I + t\partial g)^{-1}(x)$. □

³Pozri Príloha A.2 - subgradient a subdiferenciál.

⁴Pozri Príloha A.2 - Veta A.2.1.

Ďalšie poznatky o vlastnostiach a interpretáciach proximálnych operátorov, o tvaroch proximálnych operátorov pre rôzne funkcie a spôsoboch výpočtu proximálnych operátorov, ak nie sú explicitne vyjadriteľné, sú uvedené napr. v [82], [101].

2.2 Proximal-gradientná metóda (PGM)

Najznámejšia a zároveň jedna z najpoužívanějších proximálnych metód je tzv. proximal-gradientná metóda (PGM) a jej zrýchlená verzia. Informácie o proximal-gradientnej metóde možno nájsť napríklad v monografii venovanej proximálnym algoritmom [82], či v [101], [5], [4], [62], [61].

2.2.1 Formulácia a základné vlastnosti PGM

Proximal-gradientná metóda sa využíva na optimalizáciu úlohy na voľný extrém tvaru

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + g(x), \quad (2.4)$$

kde účelová funkcia $F(x)$ sa dá rozložiť na súčet dvoch funkcií $f(x)$ a $g(x)$. Funkcia $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ je diferencovateľná konvexná. Funkcia $g(x) : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ je uzavretá vlastná konvexná funkcia. Funkcia g môže byť nediferencovateľná, najlepšie taká, ktorej proximálny operátor je explicitne vyjadriteľný alebo aspoň nenáročný na výpočet pomocou špecializovaných metód.

Algoritmus základnej proximal-gradientnej metódy na riešenie úlohy (2.4) vyzerá nasledovne:

Algoritmus - PGM pre riešenie (2.4)

Vstup:	- Štartovací bod $x_0 \in \mathbb{R}^n$. - Nastavenie počítadla iterácií $k = 0$.
Výpočet:	1) Výpočet gradientu $\nabla f(x_k)$ a voľba t_k . 2) Výpočet novej iterácie $x_{k+1} = \text{prox}_{t_k g}(x_k - t_k \nabla f(x_k))$. 3) $k = k + 1$, choď na 1).

Premenná $t_k > 0$ predstavuje dĺžku kroku, ktorú volíme konštantnú alebo ju určujeme v každej iterácii. Zo zápisu algoritmu vidíme, že sa jedná o autonómnou ite-

račnú metódu (jednokrokovú), závisiacu len od bezprostredne predošlej iterácie x_k . Keďže využívame práve informáciu z gradientu funkcie, jedná sa o metódu prvého rádu. Všimnime si, že pre nulovú funkciu $g(x) \equiv 0$, dostaneme iteráciu klasickej gradientnej spádovej metódy $x_{k+1} = x_k - t_k \nabla f(x_k)$. Preto môžeme proximal-gradientnú metódu chápať ako modifikáciu gradientnej spádovej metódy umožňujúcu riešiť úlohy s nediferencovateľnými funkciami. Ak by sme chceli minimalizovať funkciu f vzhľadom na množinu C , t.j. funkcia g zo zápisu (2.4) je indikátorová funkcia $g(x) = I_C(x)$, potom je proximal-gradientná metóda zhodná s projektovanou gradientnou metódou.

V ďalšom budeme predpokladať, že optimálna funkčná hodnota F^* je konečná a funkcia ju nadobúda v bode x^* , ktorý nemusí byť jednoznačný. Konvergencia metódy je garantovaná pre voľbu konštantného kroku $t = 1/L$, čo možno sformulovať v nasledujúcej vete.

Veta 2.2.1. *Nech $f(x)$ je diferencovateľná konvexná funkcia na $\mathbb{R}^n$ s Lipschitzovsky spojitým gradientom⁵ s konštantou $L > 0$. Potom proximal-gradientná metóda s konštantným krokom*

$$t = \frac{1}{L}$$

aplikovaná na úlohu (2.4) generuje postupnosť bodov $\{x_k\}$, ktorá konverguje ako

$$F(x_k) - F^* \leq \frac{L}{2k} \|x_0 - x^*\|^2,$$

pričom bod x^ je minimom funkcie $F(x)$ a F^* je jej optimálna hodnota.*

Dôkaz. Dôkaz tvrdenia obsiahnutého vo vete možno nájsť v [101], [5], [62].

Poznámka 2.2.1. Všimnime si, že dosiahnutý výsledok je veľmi podobný výsledku pre gradientnú spádovú metódu pri podobných predpokladoch [79]. To znamená, že úlohu s nediferencovateľnou účelovou funkciou vieme riešiť s rovnakou sublineárnou rýchlosťou konvergenzie⁶ funkčných hodnôt ako v prípade úlohy s diferencovateľnou účelovou funkciou, čo je prekvapujúce. Proximal-gradientná metóda má zložitosť $O(1/\epsilon)$, čiže na dosiahnutie ϵ -presného riešenia potrebuje vykonať $O(1/\epsilon)$ krokov.

⁵Pozri Príloha A.1 - Definícia A.1.2.

⁶Pozri Príloha A.3.

Proximal-gradientná metóda je spádová metóda, hodnota účelovej funkcie sa po každej iterácii znižuje. Navyše, vzdialenosť priebežného riešenia od optimálneho sa každou iteráciou znižuje (alebo ostane rovnaká) [101], [62]. Vlastnosť spádovosti je častokrát požadovanou vlastnosťou metód, pretože vďaka nej získavame výhodu: Hodnota účelovej funkcie klesá v každej iterácii, a preto získavame v každom kroku lepšie suboptimálne riešenie (aproximáciu minima) než bolo predoslé.

2.2.2 Interpretácie PGM

Na algoritmus proximal-gradientnej metódy sa dá pozeráť mnohými spôsobmi, metóda má mnoho interpretácií [5], [82]. V nasledujúcej časti uvádzame interpretácie z [82], ktorých odvodenie sme mierne upravili pre naše budúce potreby.

Základná interpretácia PGM

Prvá, najjednoduchšia interpretácia vychádza zo stavby proximálneho operátora $prox_{tg}(x)$. Ten možno chápať ako kompromis medzi blízkosťou k bodu x a minimalizovaním funkcie g , pričom hodnotu t možno považovať za trade-off parameter. Hlavná časť algoritmu proximal-gradientnej metódy má tvar $x_{k+1} = prox_{t_k g}(x_k - t_k \nabla f(x_k))$. Preto sa táto iterácia dá chápať ako kompromis medzi štandardným gradientným krokom $x_k - t_k \nabla f(x_k)$ smerujúcim k minimu funkcie f a minimom funkcie g . Proximal-gradientnú metódu teda možno považovať za akúsi postupnú minimalizáciu, kde najprv minimalizujeme funkciu f a následne bod korigujeme, aby sme sa priblížili k minimu funkcie g .

PGM ako algoritmus minimalizácie horného ohraničenia

Tak ako v prípade gradientnej metódy a iných metód, aj proximal-gradientná metóda sa dá interpretovať ako algoritmus minimalizujúci horné ohraničenie (resp. aproximáciu účelovej funkcie) $\hat{F}_k(x)$, ktoré je konvexné a tesné v bode predchádzajúcej iterácie (t.j. funkcia spĺňa $\hat{F}_k(x) \geq F(x)$ pre každé x a $\hat{F}_k(x_k) = F(x_k)$). Algoritmy takéhoto typu iteratívne ohraničujú účelovú funkciu zhora a následne dané ohraničenie minimalizujú. Hlavná iterácia proximal-gradientnej metódy sa dá pomocou definície proximálneho

operátora (2.1) rozpísať ako

$$\begin{aligned}
 x_{k+1} &= \text{prox}_{t_k g}(x_k - t_k \nabla f(x_k)) \\
 &= \underset{u}{\operatorname{argmin}} \left(g(u) + \frac{1}{2t_k} \|u - (x_k - t_k \nabla f(x_k))\|_2^2 \right) \\
 &= \underset{u}{\operatorname{argmin}} \left(f(x_k) + \nabla f(x_k)^T (u - x_k) + \frac{1}{2t_k} \|u - x_k\|_2^2 + g(u) \right),
 \end{aligned} \tag{2.5}$$

kde posledná rovnosť bola získaná rozpísaním kvadratického člena a následnou zmenou konštantného člena. Funkcia

$$\hat{f}_k(u) := f(x_k) + \nabla f(x_k)^T (u - x_k) + \frac{1}{2t_k} \|u - x_k\|_2^2$$

je kvadratickou aproximáciou funkcie f , ktorá je tesná v bode x_k . Navyše, pre dĺžku kroku z intervalu $t_k \in (0, 1/L]$ je funkcia $\hat{f}_k$ horným ohraničením⁷ funkcie f - a teda aj funkcia $\hat{f}_k + g$ je horným ohraničením funkcie $f + g$ tesným v bode predošlej iterácie x_k . Nový bod iterácie x_{k+1} je teda bodom minima horného ohraničenia $\hat{f}_k + g$.

PGM ako metóda hľadania pevného bodu

Mnohé proximálne algoritmy môžu byť interpretované ako metódy hľadania pevných bodov vhodných operátorov [82]. V prípade proximal-gradientnej metódy hľadáme pevný bod x^* operátora z hlavnej iterácie algoritmu, t.j. bod x^* spĺňajúci

$$x^* = \text{prox}_{tg}(x^* - t \nabla f(x^*)),$$

čo sa dá použitím Vety 2.1.1. ekvivalentne prepísať na nasledujúce alternatívne tvary

$$\begin{aligned}
 x^* &= (I + t \partial g)^{-1} (I - t \nabla f)(x^*), \\
 (I - t \nabla f)(x^*) &\in (I + t \partial g)(x^*), \\
 0 &\in \nabla f(x^*) + \partial g(x^*),
 \end{aligned}$$

pričom posledný vzťah predstavuje nutnú a postačujúcu podmienkou optimality⁸ pre úlohu (2.4). Teda bod x^* minimalizuje funkciu $f + g$ práve vtedy, keď je pevným bodom operátora $(I + t \partial g)^{-1} (I - t \nabla f)$ (v literatúre známy ako forward-backward operátor [82], [101]). Minimum funkcie $f + g$ môžeme nájsť nájdením pevného bodu tohto operátora.

⁷Pozri Príloha A.1 - Veta A.1.1.

⁸Pozri Príloha A.2 - Veta A.2.1.

PGM ako diskretizačná metóda

Posledná interpretácia hovorí o prepojení proximal-gradientnej metódy a gradientného toku. Pre jednoduchosť predpokladajme, že funkcia g je diferencovateľná. Na proximal-gradientnú metódu sa možno pozeráť ako na diskretizačnú metódu pre riešenie diferenciálnej rovnice

$$\frac{d}{dt}x(t) = -\nabla f(x(t)) - \nabla g(x(t)),$$

ktorej ekvilibriá sú bodmi minima konvexnej diferencovateľnej funkcie $f + g$. Hlavná iterácia proximal-gradientnej metódy má tvar

$$x_{k+1} = \text{prox}_{t_k g}(x_k - t_k \nabla f(x_k)),$$

ktorý možno využitím Vety 2.1.1. ekvivalentne prepísať ako

$$x_{k+1} = (I + t_k \nabla g)^{-1}(I - t_k \nabla f)(x_k).$$

Z predošlého vzťahu priamo vyplýva

$$x_{k+1} + t_k \nabla g(x_{k+1}) = x_k - t_k \nabla f(x_k),$$

z čoho preusporiadaním a následným predelením dĺžkou kroku t_k dostaneme vzťah

$$\frac{x_{k+1} - x_k}{t_k} = -\nabla f(x_k) - \nabla g(x_{k+1}).$$

Poznamenajme, že v prípade nediferencovateľnej funkcie g sa predošlý zápis zmení na tvar

$$\frac{x_{k+1} - x_k}{t_k} \in -\nabla f(x_k) - \partial g(x_{k+1}). \quad (2.6)$$

Proximal-gradientnú metódu možno teda interpretovať ako diskretizačnú metódu pre riešenie diferenciálnej rovnice $\frac{d}{dt}x(t) = -\nabla f(x(t)) - \nabla g(x(t))$, ktorá používa doprednú Eulerovu diskretizáciu pre diferencovateľnú časť f a spätnú Eulerovu diskretizáciu pre (nediferencovateľnú) časť g .

Poznámka 2.2.2. Vzťah $\frac{x_{k+1} - x_k}{t_k}$ z (2.6) predstavuje smer optimalizácie pri použití proximal-gradientnej metódy. Všimnime si, že tento smer je súčtom gradientu funkcie f v súčasnom bode iterácie a subgradientu funkcie g v novom bode iterácie (súčet explicitného gradientu a implicitného subgradientu).

2.2.3 Zrýchlená proximal-gradientná metóda (APGM)

V literatúre existuje viacero pomerne príbuzných zrýchlených verzií algoritmu základnej proximal-gradientnej metódy, pozri napr. [101], [4], [78], [82]. Od základného algoritmu sa líšia v tom, že pracujú s dvomi postupnosťami bodov x_k a y_k , pričom pri výpočte bodov y_k sa využíva extrapoláčny krok. Upravený algoritmus s lepšou rýchlosťou konverencie sa nazýva zrýchlená proximal-gradientná metóda (angl. accelerated proximal-gradient method (APGM)) alebo proximal-gradientná metóda s extrapoláciou.

V práci budeme ďalej pracovať s nasledujúcou verziou zrýchlenej proximal-gradientnej metódy (APGM) z [101]. Algoritmus zrýchlenej metódy na riešenie úlohy (2.4) vyzerá nasledovne:

Algoritmus - APGM pre riešenie (2.4)

Vstup: - Štartovací bod $x_0 = y_0 \in \mathbb{R}^n$.
 - Nastavenie počítadla iterácií $k = 1$.

Výpočet: 1) Výpočet gradientu $\nabla f(y_{k-1})$ a voľba t_k .
 2) Výpočet bodu $x_k = \text{prox}_{t_k g}(y_{k-1} - t_k \nabla f(y_{k-1}))$.
 3) Extrapolácia $y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$.
 4) $k = k + 1$, choď na 1).

Tak, ako v prípade základnej proximal-gradientnej metódy, dĺžka kroku t_k býva buď konštantá alebo určovaná v každej iterácii a takisto sa jedná o iteračnú metódu prvého rádu. Zrýchlená metóda dosahuje približne rovnakú časovú náročnosť v jednej iterácii ako pôvodná proximal-gradientná metóda. Od základnej metódy sa líši v tom, že do proximálneho operátora v každej iterácii vstupujú hodnoty y_{k-1} namiesto hodnôt x_{k-1} , pričom hodnoty y_{k-1} sú získané extrapoláciou z predošlých hodnôt x_{k-1} a x_k .

Pre konvergenciu zrýchlenej proximal-gradientnej metódy musia byť splnené rovnaké predpoklady ako v prípade základnej metódy.

Veta 2.2.2. *Nech $f(x)$ je diferencovateľná konvexná funkcia na $\mathbb{R}^n$ s Lipschitzovským spojitém gradientom s konštantou $L > 0$. Potom zrýchlená proximal-gradientná metóda*

s konštantným krokom

$$t = \frac{1}{L}$$

aplikovaná na úlohu (2.4) generuje postupnosť bodov $\{x_k\}$, ktorá konverguje ako

$$F(x_k) - F^* \leq \frac{2L}{(k+1)^2} \|x_0 - x^*\|^2,$$

pričom bod x^* je minimom funkcie $F(x)$ a F^* je jej optimálna hodnota.

Dôkaz. Dôkaz tvrdenia obsiahnutého vo vete možno nájsť v [101], [5], [62].

Poznámka 2.2.3. Veta 2.2.2. hovorí o tom, že zrýchlená proximal-gradientná metóda dosahuje pri aplikovaní na úlohu (2.4) sublineárnu rýchlosť konverencie⁹ funkčných hodnôt rádu $O(\frac{1}{(k+1)^2})$. Poznamenajme, že táto rýchlosť konverencie sa za daných predpokladov nedá zlepšiť, preto je zrýchlená proximal-gradientná metóda optimálnou pre danú triedu úloh [79]. Navyše, nediferencovateľné úlohy vieme teda riešiť rovnakou rýchlosťou ako diferencovateľné. Zrýchlená proximal-gradientná metóda má zložitosť $O(1/\sqrt{\epsilon})$, t.j. potrebuje rádovo $O(1/\sqrt{\epsilon})$ iterácií na nájdenie ϵ -presného riešenia spĺňajúceho $F(x_k) - F^* \leq \epsilon$.

Zrýchlená proximal-gradientná metóda nie je spádová, nezaručuje pokles účelovej funkcie v každej iterácii tak ako základná proximal-gradientná metóda. APG metóda sa dostáva k optimálnej hodnote účelovej funkcie často pomocou tzv. „skokov“. Z uvedeného dôvodu existujú spádové verzie zrýchlenej proximal-gradientnej metódy, niektoré si spomenieme v nasledujúcej časti.

⁹Pozri Príloha A.3.

2.2.4 Spádové modifikácie APGM

Vlastnosť spádovosti je často požadovanou vlastnosťou algoritmov metód. Ukážeme si 3 modifikácie zrýchlenej proximal-gradientnej metódy, z toho druhá z nich (t.j. DPG2) je náš vlastný prístup.

V literatúre je ako spádová verzia zrýchlenej proximal-gradientnej metódy [101] označovaná metóda:

Algoritmus - Spádová DPG1 metóda pre riešenie (2.4)

Vstup: - Štartovací bod $x_0 = y_0 \in \mathbb{R}^n$.
 - Nastavenie počítadla iterácií $k = 1$.

Výpočet: 1) Výpočet gradientu $\nabla f(y_{k-1})$ a voľba t_k .
 2) $z_k = \text{prox}_{t_k g}(y_{k-1} - t_k \nabla f(y_{k-1}))$.
 3) $x_k = \begin{cases} z_k & F(z_k) \leq F(x_{k-1}) \\ x_{k-1} & \text{inak} \end{cases}$.
 4) $v_k = x_{k-1} + \frac{k+1}{2}(z_k - x_{k-1})$.
 5) Výpočet novej iterácie $y_k = (1 - \frac{2}{k+2})x_k + \frac{2}{k+2}v_k$.
 6) $k = k + 1$, choď na 1).

Ak je podmienka nerastúcosti účelovej funkcie $F(z_k) \leq F(x_{k-1})$ splnená, tak iterácia spádovej verzie algoritmu je ekvivalentná s iteráciou zrýchlenej proximal-gradientnej metódy. Ak podmienka splnená nie je, premenná x_k zostáva konštantná, až pokým sa nenájde lepšie suboptimálne riešenie.

Nasledujúcu spádovú modifikáciu sme vytvorili na základe poznatkov o základnej a zrýchlenej proximal-gradientnej metóde v nami napísanej práci [62], [61], jej praktickú implementáciu (Matlab kód) sme odvtedy však signifikantne vylepšili tak, aby šetrila výpočtový čas. Modifikácia je svojou konštrukciou podobná predošlej spádovej verzii, najväčší rozdiel je ten, že premenná x neostáva za neplatnosti podmienky konštantná, ale v každej iterácii sa približuje k optimálnemu riešeniu našej úlohy.

Algoritmus - Spádová DPG2 metóda pre riešenie (2.4)

Vstup: - Štartovací bod $x_0 = y_0 \in \mathbb{R}^n$.
 - Nastavenie počítadla iterácií $k = 1$.

Výpočet: 1) Výpočet gradientu $\nabla f(y_{k-1})$ a voľba t_k .
 2) $z_k = \text{prox}_{t_k g}(y_{k-1} - t_k \nabla f(y_{k-1}))$.
 3) $x_k = \begin{cases} z_k & F(z_k) \leq F(x_{k-1}) \\ \text{prox}_{t_k g}(x_{k-1} - t_k \nabla f(x_{k-1})) & \text{inak} \end{cases}$.
 4) Výpočet novej iterácie $y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$.
 5) $k = k + 1$, choď na 1).

Pokým je podmienka nerastúciosti účelovej funkcie $F(z_k) \leq F(x_{k-1})$ splnená, modifikácia používa zrýchlenú proximal-gradientnú metódu, kvôli jej rýchlosti konvergenzie. Ak podmienka nerastúciosti neplatí, využívame základnú proximal-gradientnú metódu, ktorá má vlastnosť spádovosti. Navyše, pri použití základnej metódy sa znižuje vzdialenosť priebežného riešenia od optimálneho, vďaka čomu priebežné riešenie x_k nestagnuje. Všimnime si, že vďaka uvedenej modifikácii robíme extrapoláčny krok v bode 4) vždy s bodom x_k , ktorý spĺňa $F(x_k) \leq F(x_{k-1})$. Takisto poznamenajme, že za zlepšenie priebežného riešenia sme „zaplatili“ výpočtom ďalšieho proximálneho operátora a gradientu $\nabla f(x_{k-1})$, čo môže byť výpočtovo náročné pre veľkorozmerné úlohy.

Tretia spádová modifikácia predstavuje zrýchlenú proximal-gradientnú metódu s reštartom metódy v prípade, ak nie je splnená podmienka nerastúciosti účelovej funkcie. Táto úprava využíva fakt, že prvá iterácia zrýchlenej proximal-gradientnej metódy sa zhoduje s iteráciou základnej metódy, čiže je spádová.

Algoritmus - APG metóda s reštartom (DPG3) pre riešenie (2.4)

Vstup: - Štartovací bod $x_0 = y_0 \in \mathbb{R}^n$.
 - Nastavenie počítadla iterácií $k = 1$.

Výpočet: 1) Výpočet gradientu $\nabla f(y_{k-1})$ a voľba t_k .
 2) Výpočet bodu $x_k = \text{prox}_{t_k g}(y_{k-1} - t_k \nabla f(y_{k-1}))$.
 3a) Ak $F(x_k) \leq F(x_{k-1})$, potom $y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$.
 3b) Inak reštart: $x_0 = y_0 = x_{k-1}$, $k = 0$.
 4) $k = k + 1$, choď na 1).

Spádová proximal-gradientná metóda s reštartom bola použitá napríklad v [52].

Poznámka 2.2.4. Účelom spádových modifikácii zrýchlenej proximal-gradientnej metódy bolo „eliminovať chybné kroky“, ktoré zhoršovali hodnotu účelovej funkcie. Prirodzene by sme očakávali rýchlejšie dosiahnutie suboptimálneho riešenia v zmysle počtu iterácií. Avšak jednoduchú hlavnú iteráciu zrýchlenej metódy sme nahradili výpočtovo zložitejšími, a preto sa toto zrýchlenie nemusí prejaviť časovo (čo je pre nás podstatnejšie). Všetky spádové verzie zrýchlenej proximal-gradientnej metódy obsahujú overovanie podmienky nerastúcosti, ktorej výpočet môže byť pre veľkorozmerné úlohy príliš drahý. Preto tieto metódy nemusia byť svojou stavbou vhodné pre optimalizáciu veľkorozmerných úloh. Uvedené spádové modifikácie s konštantou aj variabilnou dĺžkou kroku testujeme v experimentálnej časti práce.

2.3 Proximal-newtonova metóda (PNM)

Zovšeobecnením newtonovských metód (prevedením do tzv. proximálneho tvaru) tak, aby nimi bolo možné riešiť úlohy s nediferencovateľnými účelovými funkciami tvaru (2.4), vznikli tzv. proximal-newtonovské metódy [66]. Tieto metódy sa tiež dajú chápať ako zovšeobecnenie proximal-gradientnej metódy, ktoré berie do úvahy zakrivenie funkcie f . Proximal-newtonovské metódy (PNM) dosahujú za štandardných predpokladov lepšie ako lineárne rýchlosti konvergenzie [66].

2.3.1 Formulácia a základné vlastnosti PNM

Proximal-newtonova metóda využíva proximálny operátor s upravenou štruktúrou, tzv. škálovaný proximálny operátor.

Definícia 2.3.1. (Škálovaný proximálny operátor) Nech funkcia $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ je uzavretá vlastná konvexná funkcia ¹⁰ a nech H je kladne definitná matica. Škálovaný proximálny operátor $prox_g^H : \mathbb{R}^n \rightarrow \mathbb{R}^n$ funkcie g je definovaný ako:

$$prox_g^H(x) = \operatorname{argmin}_u \left(g(u) + \frac{1}{2} \|u - x\|_H^2 \right). \quad (2.7)$$

¹⁰Pozri Príloha A.1 - Definícia A.1.1.

Škálovaný proximálny operátor má niektoré vlastnosti analogické vlastnostiam štandardného proximálneho operátora. Podobne ako v prípade štandardného proximálneho operátora, škálovaný proximálny operátor pre funkciu g existuje a je jednoznačne určený pre každé $x \in \mathbb{R}^n$. Avšak použitie škálovanej normy $\|u-x\|_H$ častokrát znemožňuje explicitné vyjadrenie škálovaného proximálneho operátora.

Zo zápisu škálovaného proximálneho operátora taktiež vidíme, že ho možno interpretovať ako kompromis medzi blízkosťou k bodu x (v inej metrike ako v prípade štandardného proximálneho operátora) a minimalizovaním funkcie g .

Tak ako pri neškálovanom proximálnom operátore aj pre škálovaný existuje prepojenie so subdiferenciálom funkcie g . Škálovaný proximálny operátor $prox_g^H$ a subdiferenciálny operátor ∂g sú prepojené vzťahom

$$prox_g^H = (I + H^{-1}\partial g)^{-1}. \quad (2.8)$$

Uvedený vzťah sa dá odvodiť analogicky ako v prípade Vety 2.1.1..

Algoritmus proximal-newtonovej metódy na riešenie úlohy (2.4) má podobnú štruktúru ako algoritmus proximal-gradientnej metódy. Najväčší rozdiel je ten, že bod 2) v algoritme PGM nahradíme - základná iteračná schéma proximal-newtonovej metódy vyzerá nasledovne

$$x_{k+1} = prox_{t_k g}^{H_k}(x_k - t_k H_k^{-1} \nabla f(x_k)), \quad (2.9)$$

kde premenná t_k je dĺžka kroku a matica H_k je Hessova matica $\nabla^2 f(x_k)$ alebo symetrická kladne definitná aproximácia Hessovej matice funkcie f , t.j. $H_k \approx \nabla^2 f(x_k)$.

2.3.2 Interpretácie PNM

Na hlavnú časť algoritmu proximal-newtonovej metódy (2.9) sa dá pozeráť mnohými spôsobmi, metóda má viacero interpretácií. Pozrime sa na štyri z nich ako pri proximal-gradientnej metóde. Na základe rôznych interpretácií PGM z [82] sme urobili analogické interpretácie pre PNM. Pre zachovanie čitateľnosti vynechajme indexy $\cdot_k$ a označme nový bod iterácie $x^+ = x_{k+1}$.

Hlavná časť algoritmu proximal-newtonovej metódy potom vyzerá nasledovne

$$x^+ = prox_{t_g}^H(x - tH^{-1}\nabla f(x)). \quad (2.10)$$

Základná interpretácia PNM

Prvá najjednoduchšia interpretácia vychádza priamo z definície škálovaného proximálneho operátora $\text{prox}_{tg}^H(x)$. Hlavnú časť algoritmu (2.10) môžeme teda chápať ako kompromis medzi minimalizáciou funkcie g a blízkosťou (v reči škálovanej normy) k newtonovskému kroku $x - tH^{-1}\nabla f(x)$ pre minimalizáciu funkcie f . Proximal-newtonovu metódu teda možno považovať za akúsi postupnú minimalizáciu, kde bodom najprv pohneme v smere minimalizujúcom funkciu f a následne bod korigujeme, aby sme sa zároveň priblížili minimu funkcie g .

PNM ako algoritmus minimalizácie aproximácie účelovej funkcie

Tak ako v prípade proximal-gradientnej metódy aj na proximal-newtonovu metódu sa dá pozeráť ako na algoritmus, ktorý iteratívne aproximuje účelovú funkciu a následne danú aproximáciu minimalizuje. Ak rozpíšeme hlavnú iteráciu proximal-newtonovej metódy (2.10) pomocou definície škálovaného proximálneho operátora (2.7), dostaneme

$$\begin{aligned} x^+ &= \operatorname{argmin}_u \left(g(u) + \frac{1}{2t} \|(u - x) + tH^{-1}\nabla f(x)\|_H^2 \right) \\ &= \operatorname{argmin}_u \left(f(x) + \nabla f(x)^T(u - x) + \frac{1}{2t}(u - x)^T H(u - x) + g(u) \right), \end{aligned} \quad (2.11)$$

kde posledná rovnosť bola získaná rozpísaním kvadratického člena a následnou výmenou konštantného člena. Funkcia

$$\hat{f}(u) := f(x) + \nabla f(x)^T(u - x) + \frac{1}{2t}\|u - x\|_H^2$$

je aproximáciou funkcie f druhého rádu, ktorá je tesná v bode x . Nový bod iterácie x^+ je teda minimom funkcie $\hat{f} + g$, ktorá je aproximáciou účelovej funkcie $f + g$.

Poznámka 2.3.1. Spomínali sme, že škálovaný proximálny operátor sa väčšinou nedá dobre explicitne vyjadriť. Naopak, jeho vyčísľovanie býva výpočtovo extrémne náročné. Z toho dôvodu sa v proximal-newtonovej metóde v praxi nový bod iterácie počíta riešením podúlohy (2.11). Túto podúlohu treba riešiť inými iteračnými metódami, napr. autori publikácie [66] používajú súradnicové spádové metódy. V prípade úlohy l_1 -regulovaných najmenších štvorcov (1.2) však táto podúloha predstavuje podobne zložitý problém, ako je pôvodná úloha. Z uvedených dôvodov sa proximal-newtonovej metóde v experimentálnej časti práce venovať nebudeme.

PNM ako metóda hľadania pevného bodu

Ďalšou interpretáciou proximal-newtonovej metódy je, že ju môžeme považovať za metódu hľadania pevného bodu operátora z hlavnej iterácie (2.10). Pevný bod x^* teda spĺňa

$$x^* = \text{prox}_{tg}^H(x^* - tH^{-1}\nabla f(x^*)),$$

čo sa dá pomocou vzťahu (2.8) prepísať na nasledujúce ekvivalentné tvary

$$\begin{aligned} x^* &= (I + tH^{-1}\partial g)^{-1}(I - tH^{-1}\nabla f)(x^*), \\ (I - tH^{-1}\nabla f)(x^*) &\in (I + tH^{-1}\partial g)(x^*), \\ -tH^{-1}\nabla f(x^*) &\in tH^{-1}\partial g(x^*), \\ 0 &\in \nabla f(x^*) + \partial g(x^*), \end{aligned}$$

pričom posledný vzťah predstavuje nutnú a postačujúcu podmienkou optimality pre úlohu (2.4). Minimum funkcie $f + g$ môžeme nájsť nájdením pevného bodu operátora $(I + tH^{-1}\partial g)^{-1}(I - tH^{-1}\nabla f)$ z hlavnej iterácie (2.10).

PNM ako diskretizačná metóda

Proximal-newtonovu metódu taktiež možno považovať za diskretizačnú metódu pre riešenie vhodnej diferenciálnej rovnice, ktorej ekvilibriá sú bodmi minima konvexnej funkcie $f + g$. Pre jednoduchosť predpokladajme, že funkcia g je diferencovateľná. Hlavnú iteráciu proximal-newtonovej metódy (2.10) možno využitím vlastnosti (2.8) ekvivalentne prepísať na

$$x^+ = (I + tH^{-1}\nabla g)^{-1}(I - tH^{-1}\nabla f)(x),$$

čo je ekvivalentné vzťahu

$$x^+ + tH^{-1}\nabla g(x^+) = x - tH^{-1}\nabla f(x).$$

Preusporiadaním predošlej rovnosti a následným jej predelením parametrom t dostaneme vzťah

$$\frac{x^+ - x}{t} = -H^{-1}(\nabla f(x) + \nabla g(x^+)).$$

V prípade nediferencovateľnej funkcie g sa predošlý zápis zmení na tvar

$$\frac{x^+ - x}{t} \in -H^{-1}(\nabla f(x) + \partial g(x^+)). \quad (2.12)$$

Proximal-newtonova metóda predstavuje diskretizačnú metódu pre riešenie diferenciálnej rovnice $\frac{d}{dt}x(t) = -H^{-1}(\nabla f(x(t)) + \nabla g(x(t)))$, ktorá používa doprednú Eulerovu diskretizáciu pre diferencovateľnú časť f a spätnú Eulerovu diskretizáciu pre (nediferencovateľnú) časť g .

Poznámka 2.3.2. Vzťah $\frac{x^+ - x}{t}$ z (2.12) predstavuje smer optimalizácie pri použití proximal-newtonovej metódy. Tento smer je newtonovského typu, pričom kombinuje explicitný gradient a implicitný subgradient.

2.4 Nová proximal-kvázinewtonova metóda (PQNM)

Táto časť prezentuje náš pokus vytvoriť proximal-kvázinewtonovskú metódu (PQNM). Presnejšie ide o snahu skombinovať základnú proximal-gradientnú metódu (resp. štandardný proximálny operátor) a kvázinewtonovskú metódu ako napríklad BFGS či DFP.

2.4.1 Motivácia a popis PQNM

Medzi hlavné motivácie na vytvorenie proximal-kvázinewtonovej metódy patrí snaha zrýchliť konvergenciu proximal-gradientnej metódy použitím kvázinewtonovskej metódy a takisto ponechanie štandardného proximálneho operátora (2.1), ktorý má pre mnohé funkcie explicitný tvar, resp. ktorého hodnotu vieme pre mnohé funkcie efektívne vyčíslovať jednoduchými špecializovanými metódami (toto pri proximal-newtonovej metóde a škálovanom proximálnom operátore neplatí). Navyše využitím kvázinewtonovských metód odpadá potreba riešiť sústavu n lineárnych rovníc, pretože môžeme aproximovať priamo inverznú Hessovú maticu.

Algoritmus navrhovanej proximal-kvázinewtonovej metódy na riešenie úlohy na voľný extrém (2.4) má príbuznú štruktúru ako algoritmus proximal-gradientnej metódy. Hlavnou zmenou je, že bod 2) v algoritme proximal-gradientnej metódy nahradíme iteračným predpisom

$$x_{k+1} = \text{prox}_{t_k g}(x_k - t_k B_k \nabla f(x_k)), \quad (2.13)$$

kde premenná t_k je dĺžka kroku a matica B_k je symetrická kladne definitná aproximácia inverznej Hessovej matice funkcie f v bode x_k , t.j. $B_k \approx [\nabla^2 f(x_k)]^{-1}$. Všimnime si, že

v zápise vystupuje štandardný neškálovaný proximálny operátor.

2.4.2 Interpretácie PQNM

Po nie príliš presvedčivých počítačových numerických experimentoch s proximal-kvázinewtonovou metódou, sme sa rozhodli pozrieť na teoretické interpretácie/vlastnosti uvedenej metódy. Na hlavnú časť algoritmu proximal-kvázinewtonovej metódy (2.13) sa dá pozeráť viacerými spôsobmi. V nasledujúcej časti odvodíme 4 pohľady na hlavnú iteráciu metódy tak, ako to bolo urobené v [82] pre prípad proximal-gradientnej metódy a v odseku 2.3.2 pre prípad PNM. Pre zachovanie čitateľnosti vynechajme indexy $\cdot_k$ a označme nový bod iterácie $x^+ = x_{k+1}$.

Hlavná časť algoritmu navrhovanej proximal-kvázinewtonovej metódy po tomto zjednodušení zápisu vyzerá

$$x^+ = \text{prox}_{tg}(x - tB\nabla f(x)). \quad (2.14)$$

Základná interpretácia PQNM

Prvá najjednoduchšia interpretácia metódy vychádza priamo z definície štandardného proximálneho operátora (2.1). Nový bod iterácie (2.14) môžeme teda chápať ako kompromis medzi minimalizáciou funkcie g a blízkosťou ku štandardnému kvázinewtonovskému kroku $x - tB\nabla f(x)$ pre minimalizáciu funkcie f , pričom premenná t predstavuje trade-off parameter. Proximal-kvázinewtonovu metódu teda možno považovať za akúsi postupnú minimalizáciu, kde najprv minimalizujeme funkciu f a následne získaný bod korigujeme, aby sme sa priblížili minimu funkcie g .

PQNM ako algoritmus minimalizácie aproximácie účelovej funkcie

Tak ako v prípade predošlých proximálnych metód aj na proximal-kvázinewtonovu metódu sa dá pozeráť ako na algoritmus, ktorý sa iteratívne snaží aproximovať účelovú funkciu a následne danú aproximáciu minimalizuje. Ak rozpíšeme hlavnú iteráciu proximal-kvázinewtonovej metódy (2.14) pomocou definície proximálneho operátora

(2.1), dostaneme

$$\begin{aligned} x^+ &= \operatorname{argmin}_u \left(g(u) + \frac{1}{2t} \|(u - x) + tB\nabla f(x)\|_2^2 \right) \\ &= \operatorname{argmin}_u \left(f(x) + \nabla f(x)^T B(u - x) + \frac{1}{2t} \|u - x\|_2^2 + g(u) \right), \end{aligned} \quad (2.15)$$

kde posledná rovnosť bola získaná rozpísaním kvadratického člena a následnou zmenou konštantného člena. Nový bod iterácie x^+ je teda minimom funkcie $\hat{f} + g$, kde

$$\hat{f}(u) := f(x) + \nabla f(x)^T B(u - x) + \frac{1}{2t} \|u - x\|_2^2.$$

Funkciu f z úlohy (2.4) sa teda algoritmus snaží aproximovať funkciou $\hat{f}$, ktorá je tesná v bode x (platí $\hat{f}(x) = f(x)$). Avšak táto funkcia nemusí predstavovať dobrú aproximáciu funkcie f .

PQNM ako metóda hľadania pevného bodu

Pozrime sa teraz, čo bude splňať pevný bod x^* operátora z hlavnej iterácie proximal-kvázinewtonovej metódy (2.14). Zaujíma nás teda vzťah

$$x^* = \operatorname{prox}_{tg}(x^* - tB\nabla f(x^*)),$$

čo sa dá využitím Vety 2.1.1. upraviť na nasledujúce ekvivalentné tvary

$$\begin{aligned} x^* &= (I + t\partial g)^{-1}(I - tB\nabla f)(x^*), \\ (I - tB\nabla f)(x^*) &\in (I + t\partial g)(x^*), \\ -tB\nabla f(x^*) &\in t\partial g(x^*), \\ 0 &\in B\nabla f(x^*) + \partial g(x^*). \end{aligned}$$

Všimnime si, že kvôli výskytu matice B posledný vzťah nepredstavuje nutnú a postačujúcu podmienkou optimality pre úlohu (2.4), ako by sme požadovali.

PQNM ako diskretizačná metóda

V nasledujúcom pre jednoduchosť predpokladajme, že funkcia g je diferencovateľná. Tak ako ostatné proximálne metódy, aj proximal-kvázinewtonovu metódu možno považovať za diskretizačnú metódu pre riešenie nejakej diferenciálnej rovnice. Hlavnú

iteráciu proximal-kvázinevtonovej metódy (2.14) možno použitím Vety 2.1.1. prepísať na

$$x^+ = (I + t\nabla g)^{-1}(I - tB\nabla f)(x),$$

čo je ekvivalentné vzťahu

$$x^+ + t\nabla g(x^+) = x - tB\nabla f(x).$$

Preusporiadaním predošlej rovnosti a jej následným predelením parametrom t dostaneme vzťah

$$\frac{x^+ - x}{t} = -B\nabla f(x) - \nabla g(x^+).$$

Proximal-kvázinevtonova metóda predstavuje diskretizačnú metódu pre riešenie diferenciálnej rovnice $\frac{d}{dt}x(t) = -B\nabla f(x(t)) - \nabla g(x(t))$, ktorá používa doprednú diskretizáciu pre diferencovateľnú časť f a spätnú diskretizáciu pre (nediferencovateľnú) časť g . Takisto vzťah $\frac{x^+ - x}{t}$ predstavuje smer optimalizácie pri použití navrhovanej proximal-kvázinevtonovej metódy.

Výsledky z predošlých možných interpretácii proximal-kvázinevtonovej metódy (posledné 3 okrem prvej najjednoduchšej) nám napovedajú, že táto nami navrhnutá metóda v skutočnosti neminimalizuje účelovú funkciu $f + g$ z úlohy (2.4) ako sme pôvodne zamýšľali. Prehľadný súhrn týchto odvodených teoretických vlastností pre PGM, PNM a PQNM metódy uvádzame v Tabuľke 2.2..

Metóda	PGM	PNM	PQNM
Nový bod iterácie	$x^+ = \text{prox}_{t_g}(x - t\nabla f(x))$	$x^+ = \text{prox}_{t_g^H}(x - tH^{-1}\nabla f(x))$	$x^+ = \text{prox}_{t_g}(x - tB\nabla f(x))$
Aproximácia funkcie $f(u)$ v $x^+ = \text{argmin}_u(\hat{f}(u) + g(u))$	$\hat{f}(u) = f(x) + \nabla f(x)^T(u - x) + \frac{1}{2t}\ u - x\ _2^2$	$\hat{f}(u) = f(x) + \nabla f(x)^T(u - x) + \frac{1}{2t}\ u - x\ _H^2$	$\hat{f}(u) = f(x) + \nabla f(x)^TB(u - x) + \frac{1}{2t}\ u - x\ _2^2$
Pevný bod x^* spĺňa	$0 \in \nabla f(x^*) + \partial g(x^*)$	$0 \in \nabla f(x^*) + \partial g(x^*)$	$0 \in B\nabla f(x^*) + \partial g(x^*)$
Smer optimalizácie $\frac{x^+ - x}{t}$	$\in -\nabla f(x) - \partial g(x^+)$	$\in -H^{-1}(\nabla f(x) + \partial g(x^+))$	$\in -B\nabla f(x) - \partial g(x^+)$

Tabuľka 2.2: Súhrn teoretických vlastností pre proximal-gradientnú, proximal-newtonovu a novú proximal-kvázineutronovu metódu.

2.5 Prírastková proximal-gradientná metóda (PPGM)

Pre minimalizáciu účelovej funkcie v tvare priemeru (súčtu) funkcií vznikli, tzv. prírastkové metódy (angl. incremental methods) [10], [11], ktoré pozostávajú z výpočtovo jednoduchších iterácií aplikovaných na jednotlivé komponenty účelovej funkcie. Publikácia [10] predstavuje zjednotený pohľad na prírastkové metódy, autori v [10] analyzujú konvergenčné vlastnosti prírastkových metód, ktoré kombinujú subgradientné a proximálne kroky. Na základe druhej kapitoly v [10] uvádzame nasledujúcu prírastkovú proximal-gradientnú metódu (PPGM), ktorá sa dá chápať ako podprípád metódy z [10] a ktorú sme upravili, aby bola použiteľná na úlohy nami požadovaného tvaru.

2.5.1 Popis a základné vlastnosti PPGM

Prírastková proximal-gradientná metóda sa využíva na optimalizáciu úlohy na voľný extrém tvaru

$$\min_{x \in \mathbb{R}^n} F(x) := \frac{1}{m} \sum_{i=1}^m f_i(x) + g(x), \quad (2.16)$$

kde funkcie $f_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ sú diferencovateľné konvexné. Funkcia $g(x) : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ je uzavretá vlastná konvexná funkcia. Funkcia g môže byť nediferencovateľná, avšak pre efektívne použitie metódy požadujeme, aby mala explicitne vyjadriteľný alebo výpočtovo nenáročný proximálny operátor.

V nasledujúcom uvádzame algoritmus prírastkovej proximal-gradientnej metódy na riešenie úlohy (2.16):

Algoritmus - PPGM pre riešenie (2.16)

- Vstup:**
- Štartovací bod $x_0 \in \mathbb{R}^n$.
 - Nastavenie počítadla iterácií $k = 0$.
- Výpočet:**
- 1) Zvoľ index $i_k = (k \bmod m) + 1$.
 - 2) Výpočet gradientu $\nabla f_{i_k}(x_k)$ a voľba t_k .
 - 3) Výpočet novej iterácie $x_{k+1} = \text{prox}_{t_k g/m}(x_k - \frac{t_k}{m} \nabla f_{i_k}(x_k))$.
 - 4) $k = k + 1$, choď na 1).

Všimnime si, že prírastková proximal-gradientná metóda pracuje v každej iterácii iba s jedným komponentom f_i a nie s celým súčtom funkcií $\sum_{i=1}^m f_i(x)$ (ako proximal-

gradientná metóda). Výpočet čiastkového gradientu ∇f_i je m -násobne jednoduchší ako výpočet plného gradientu $\sum_{i=1}^m \nabla f_i(x)$. Preto má prírastková proximal-gradientná metóda najväčší zmysel, ak je počet komponentov m veľký. Jednotlivé funkcie f_i vyberáme postupne cyklicky, index i_k sa v každej iterácii mení. V súčasnosti prebiehajú snahy o skúmanie stochastických metód (napr. [60], [92], [76], [27]), pri ktorých sa index i_k volí náhodne, častokrát z rovnomerného rozdelenia. Stochastické metódy často dosahujú lepšie konvergenčné výsledky avšak, kvôli náhodnosti, v strednej hodnote. V porovnaní s ich deterministickými verziami dosahujú menšiu odchýlku od optimálnej hodnoty účelovej funkcie za rovnaký očakávaný (stredná hodnota) počet iterácii [10]. Stochastické verzie algoritmov častokrát povoluju použitie väčších krokov pre ich konvergenciu ako deterministické, napr. [92], [10]. Poznamenajme však, že porovnávanie „worst-case“ hodnôt (deterministické metódy) s hodnotami v strednej hodnote (stochastické metódy) nemusí byť presne platné. Iná modifikácia prírastkových metód, ktorou by sa dalo vyhnúť nevhodnému usporiadaniu čiastkových funkcií f_i , spočíva v permutovaní funkcií f_i po každých m iteráciach. Poznamenajme, že prírastková proximal-gradientná metóda po m iteráciach sa dá interpretovať ako proximal-gradientná metóda s nepresným smerom optimalizácie (obsahuje chybový člen).

Premenná $t_k > 0$ predstavuje dĺžku kroku, ktorú volíme konštantnú alebo ju upravujeme v každej iterácii. Konštantná dĺžka kroku (preferovaná forma implementácie metódy) však negarantuje konvergenciu, prírastková proximal-gradientná metóda typicky osciluje v okolí riešenia úlohy (2.16), pričom veľkosť oscilácie je zhruba proporcionálna k t/m [10]. Pre zmenšujúci sa krok $t_k = O(1/k)$ metóda konverguje pomalou sublineárnou rýchlosťou [10].

Ak by sme chceli porovnať proximal-gradientnú metódu s prírastkovou proximal-gradientnou metódou, tak porovnanie môžeme rozdeliť na dve časti. V prípade, keď sme ďaleko od bodu optima úlohy (2.16), prírastková proximal-gradientná metóda bude konvergovať rýchlejšie - ak jednotlivé funkcie f_i nie sú príliš odlišné, tak ďaleko od optima je gradient ∇f_i viacmenej správnym smerom [10]. Naopak v prípade, keď sme už blízko optima, štandardná proximal-gradientná metóda je zvyčajne lepšia kvôli pomalejšej konvergencii prírastkovej metódy.

2.5.2 Modifikácie prírastkovej proximal-gradientnej metódy

Z predošlých poznatkov vyplýva, že za hlavné nevýhody prírastkovej proximal-gradientnej metódy možno považovať pomalú rýchlosť konverencie a fakt, že použitie konštantných dĺžkok krokov negarantuje konvergenciu. Z toho dôvodu uvádzame dve modifikácie či rozšírenia prírastkovej proximal-gradientnej metódy, ktoré by mali uvedené nevýhody eliminovať.

Prvú modifikáciu predstavuje metóda inšpirovaná prírastkovou metódou IAG (Incremental Aggregated Gradient) z [13], resp. deterministickou verziou stochastickej SAG (stochastic average gradient) metódy z [92]. IAG metódu sme upravili do proximálneho tvaru (využíva proximálny operátor), aby sme vedeli optimalizovať nediferencovateľné funkcie tvaru (2.16). Túto vzniknutú prírastkovú metódu budeme v ďalšom texte nazývať PIAG metóda.

Algoritmus PIAG metódy na riešenie úlohy (2.16) má nasledovný tvar:

Algoritmus - PIAG metóda pre riešenie (2.16)

Vstup:

- Štartovací bod $x_0 \in \mathbb{R}^n$.
- Nastavenie počítadla iterácií $k = 0$.

Výpočet:

- 1) Zvoľ index $i_k = (k \bmod m) + 1$.
- 2) Výpočet gradientu $\nabla f_{i_k}(x_k)$ a voľba t_k .
- 3) Update $y_{i,k} = \begin{cases} \nabla f_{i_k}(x_k) & i = i_k \\ y_{i,k-1} & \text{inak} \end{cases}$.
- 4) Výpočet novej iterácie $x_{k+1} = \text{prox}_{t_k g}(x_k - \frac{t_k}{m} \sum_{i=1}^m y_{i,k})$.
- 5) $k = k + 1$, choď na 1).

Druhú modifikáciu predstavuje deterministická verzia MISO (Minimization by Incremental Surrogate Optimization) metódy z [76], [75], ktorú sme pre lepšiu interpretáciu a porovnanie s predošlými metódami prepísali do ekvivalentného tvaru, v ktorom vystupuje proximálny operátor.

Algoritmus MISO metódy na riešenie úlohy (2.16) vyzerá nasledovne:

Algoritmus - MISO metóda pre riešenie (2.16)

Vstup: - Štartovací bod $x_0 \in \mathbb{R}^n$.
 - Nastavenie počítadla iterácií $k = 0$.

Výpočet: 1) Zvoľ index $i_k = (k \bmod m) + 1$.
 2) Výpočet gradientu $\nabla f_{i_k}(x_k)$ a voľba t_k .
 3) Update $y_{i,k} = \begin{cases} x_k - t_k \nabla f_{i_k}(x_k) & i = i_k \\ y_{i,k-1} & \text{inak} \end{cases}$.
 4) Výpočet novej iterácie $x_{k+1} = \text{prox}_{t_k g}(\frac{1}{m} \sum_{i=1}^m y_{i,k})$.
 5) $k = k + 1$, choď na 1).

PIAG aj MISO iterácia je podobne výpočtovo nenáročná ako v prípade prírastkovej proximal-gradientnej metódy, v každom kroku sa vyčísľuje len jeden čiastkový gradient $\nabla f_{i_k}(x_k)$. Vidíme, že metódy PIAG a MISO sú svojou myšlienkou veľmi príbuzné, v proximálnom operátore PIAG priemeruje predošlé najaktuálnejšie (pre daný index i) čiastkové gradienty a MISO priemeruje predošlé najaktuálnejšie gradientné kroky. PIAG a MISO metódy sú pomerne pamäťovo náročné, pretože obe na svoj chod potrebujú v pamäti držať m najaktuálnejších hodnôt čiastkových gradientov (resp. predošlých gradientných krokov). Na rozdiel od prírastkovej proximal-gradientnej metódy, každá iterácia PIAG (MISO) metódy využíva všetky gradienty funkcií f_i (podobne ako v prípade štandardnej proximal-gradientnej metódy), preto je smer optimalizácie presnejší a menej náchylný na prudké zmeny ako v prípade prírastkovej proximal-gradientnej metódy. Pôvodná SAG metóda dosahuje pre úlohy so silnokonvexnou (konvexnou) účelovou funkciou lineárnu (sublineárnu $O(1/k)$) rýchlosť konvergenzie¹¹ a podporuje konštantné dĺžky krokov [92]. Podobné výsledky dosahuje aj MISO metóda [76]. V literatúre existujú aj ďalšie metódy s podobnou myšlienkou, napr. [27].

2.6 ADMM metóda

ADMM (angl. Alternating Direction Method of Multipliers) je proximálna metóda, v literatúre taktiež známa ako Douglas-Rachford splitting [16], [82]. Publikácia [16] je monografiou, ktorá sa venuje ADMM metóde.

¹¹Pozri Príloha A.3.

2.6.1 Popis a základné vlastnosti ADMM

ADMM metóda sa používa na optimalizáciu úlohy na voľný extrém tvaru

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + g(x), \quad (2.17)$$

kde funkcie $f(x), g(x) : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ sú obe uzavreté vlastné konvexné funkcie. Funkcie f a g môžu byť nediferencovateľné, najlepšie také, ktorých proximálny operátor je explicitne vyjadriteľný alebo výpočtovo nenáročný. Hlavná výhoda ADMM metódy a zároveň najväčší rozdiel metódy oproti predošlým metódam spočíva v tom, že k častiam účelovej funkcie pristupuje oddelene - iba pomocou ich proximálnych operátorov.

Algoritmus ADMM metódy na riešenie úlohy (2.17) pracuje s tromi postupnosťami bodov x_k, z_k a u_k a vyzerá nasledovne:

Algoritmus - ADMM pre riešenie (2.17)

Vstup:	- Štartovací bod $z_0 = u_0 \in \mathbb{R}^n$. - Nastavenie počítadla iterácií $k = 0$. - Dĺžka kroku t.
Výpočet:	1) Výpočet bodu $x_{k+1} = \text{prox}_{tf}(z_k - u_k)$. 2) Výpočet bodu $z_{k+1} = \text{prox}_{tg}(x_{k+1} + u_k)$. 3) Výpočet bodu $u_{k+1} = u_k + x_{k+1} - z_{k+1}$. 4) $k = k + 1$, choď na 1).

Poznámka 2.6.1. Podľa [82] postupnosti bodov x_k a z_k obe konvergujú k optimálnemu riešeniu úlohy (2.17) a tým pádom aj k sebe navzájom. Avšak z konštrukcie ADMM algoritmu a vlastností proximálneho operátora (2.1) vyplýva, že tieto postupnosti majú odlišné vlastnosti. Presnejšie, postupnosť bodov x_k pochádza vždy z definičného oboru funkcie f , a naopak postupnosť bodov z_k je z definičného oboru funkcie g . Preto v prípade, ak by funkcia g kodovala ohraňovania úlohy, body z_k budú generované ako prípustné, zatiaľ čo body x_k môžu byť neprípustné. Podobne, ak proximálny operátor pre funkciu g spôsobuje nejakú špeciálnu vlastnosť (napríklad ak $g(x) = \|x\|_1$, tak proximálny operátor prox_{tg} spôsobuje riedkosť), túto vlastnosť budú spĺňať vždy akurát body z_k .

ADMM metóda konverguje už za pomerne slabých predpokladoch, zložitosť metódy v najhoršom prípade je $O(1/\epsilon^2)$ [16], čo je slabý výsledok. Metóda však napriek tomu v praxi funguje častokrát efektívne. Vďaka faktu, že ADMM pristupuje k funkciám f a g zvlášť pomocou proximálnych operátorov, pre plne separovateľné funkcie $f(x) = \sum_{i=1}^n f_i(x_i)$ je tento algoritmus veľmi dobre paralelizovateľný. Takisto sa dá algoritmus dobre prispôbiť a paralelizovať aj v prípade $f(x) = \sum_{i=1}^n f_i(x)$ [16].

2.6.2 Interpretácia ADMM metódy

Tak ako v prípade predošlých proximálnych metód, aj v prípade ADMM metódy existujú viaceré interpretácie algoritmu. Metóda sa dá interpretovať napríklad ako postupná minimalizácia rozšírenej Lagrangeovej funkcie, takisto ako diskretizačná metóda na riešenie systému obyčajných diferenciálnych rovníc [82].

ADMM ako metóda hľadania pevného bodu

Jednou možnou interpretáciou ADMM metódy je, že ju môžeme považovať za metódu hľadania pevného bodu hlavnej iterácie, ktorý potom spĺňa nutné a postačujúce podmienky optimality pre úlohu (2.17). Pevné body x^* , z^* , u^* teda spĺňajú

$$x^* = \text{prox}_{tf}(z^* - u^*),$$

$$z^* = \text{prox}_{tg}(x^* + u^*),$$

$$u^* = u^* + x^* - z^*,$$

kde z poslednej rovnosti vyplýva $x^* = z^*$. Eliminovaním premennej z^* a použitím Vety 2.1.1. sa prvé dva vzťahy upravajú na

$$x^* = (I + t\partial f)^{-1}(x^* - u^*),$$

$$x^* = (I + t\partial g)^{-1}(x^* + u^*).$$

Ekvivalentne sa tieto vzťahy dajú prepísať na

$$-u^* \in t\partial f(x^*), \quad u^* \in t\partial g(x^*).$$

Ak tieto vzťahy sčítame, vidíme, že pevný bod x^* (a teda aj bod z^*) spĺňa nutnú a postačujúcu podmienku optimality $0 \in \partial f(x^*) + \partial g(x^*)$ pre úlohu (2.17).

Kapitola 3

Metódy vnútorného bodu

V súčasnosti sú metódy vnútorného bodu (MVB) najpoužívanějšími metódami na riešenie diferencovateľných konvexných optimalizačných úloh. V tejto časti sa budeme venovať hlavne tzv. primárno-duálnym metódam vnútorného bodu pre riešenie všeobecných úloh konvexnej optimalizácie. Viac informácií o primárno-duálnych MVB možno nájsť napr. v [14], [105], [59], [63], [61], [64] a o bariérových MVB napr. v [14], [80], [48].

3.1 Formulácia úlohy a základná myšlienka MVB

Uvažujme nasledujúcu všeobecnú úlohu konvexného programovania

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m, \end{aligned} \tag{3.1}$$

kde $f_0, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$ sú konvexné, dvakrát spojitاً diferencovateľné funkcie. Konvexnosť funkcií f_i z ohraničení (t.j. $i = 1, \dots, m$) implikuje konvexnosť množiny prípustných riešení

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

Úlohou je teda nájsť minimum konvexnej účelovej funkcie $f_0(x)$ na konvexnej množine prípustných riešení P .

O úlohe (3.1) predpokladáme, že je riešiteľná, t.j. existuje jej optimálne riešenie x^* . Tiež predpokladáme, že existuje tzv. vnútorný bod množiny prípustných riešení, t.j.

bod $x \in P$ taký, ktorý spĺňa $f_i(x) < 0$, $i = 1, \dots, m$. Tento predpoklad znamená, že vnútro množiny prípustných riešení P je neprázdne, a teda je splnená Slaterova podmienka (úloha (3.1) je S-regulárna).

Základnou myšlienkou bariérových metód vnútorného bodu je riešiť pôvodnú úlohu (3.1) pomocou riešenia postupnosti parametrizovaných úloh na voľný extrém pre zväčšujúcu sa hodnotu parametra, pričom riešenie aktuálnej úlohy je použité ako štartovací bod pre ďalšiu minimalizáciu. Postupnosť miním týchto parametrizovaných úloh konverguje k optimálnemu riešeniu pôvodnej úlohy (3.1) a tvorí tzv. centrálnu trajektóriu.

Primárno-duálne metódy vnútorného bodu, na rozdiel od bariérových metód, parametrizujú a riešia nutné a postačujúce podmienky optimality. Hlavnou myšlienkou primárno-duálnych metód vnútorného bodu je, že podmienka komplementarity v nutných a postačujúcich podmienkach optimality sa perturbuje kladným parametrom $\frac{1}{t}$. Tým vznikne postupnosť perturbovaných systémov rovníc, ktoré sa riešia Newtonovou metódou pre postupne zväčšujúcu sa hodnotu parametra t . Postupnosť riešení týchto systémov konverguje k optimálnemu riešeniu pôvodnej úlohy (a leží na tzv. centrálnej trajektórii).

Poznamenajme, že tieto parametrizované podúlohy (parametrizované úlohy na voľný extrém či parametrizované systémy rovníc) pri MVB nie je potrebné riešiť presne. Hovorí sa o tzv. voľnom sledovaní centrálnej trajektórie v jej určitom vymedzenom okolí, ktoré využíva kvadratickú konvergenciu¹ Newtonovej metódy v blízkosti minima. Metódy vnútorného bodu sa teda spoliehajú na Newtonovu metódu pri počítaní smerov optimalizácie, a tým pádom sú metódami druhého rádu zdieľajúc ich výhody a nevýhody.

3.2 Primárno-duálne metódy vnútorného bodu

Keďže v experimentálnej časti práce budeme využívať z MVB iba primárno-duálne algoritmy, vysvetlíme teraz bližšie princípy ich fungovania. Vychádzame predovšetkým z [14], pričom sme tam uvedené postupy a algoritmus upravili pre potreby úlohy (3.1).

Podľa [59] primárno-duálne metódy vnútorného bodu (PDMVB) majú najlepšie

¹Pozri Príloha A.3.

teoretické konvergenčné vlastnosti spomedzi viacerých metód vnútorného bodu, a podľa [44], [105] majú takisto aj najlepšiu praktickú konvergenciu. Metódy vnútorného bodu pre úlohy lineárneho a kvadratického programovania vedia vyriešiť veľkorozmené úlohy za desiatky iterácií.

Výhodou primárno-duálnej metódy je, že v každej iterácii sú aktualizované primárne a zároveň duálne premenné. Napriek rozdielom sú primárno-duálne metódy príbuzné bariérovým metódam - primárno-duálne smery optimalizácie sú svojou štruktúrou veľmi podobné smerom optimalizácie bariérovej metódy [14].

3.2.1 Primárno-duálny smer v MVB

Základ primárno-duálnej metódy vnútorného bodu tvorí tzv. hľadací smer (angl. search direction), ktorý predstavuje smer optimalizácie primárnych a duálnych premenných. Získava sa aplikovaním Newtonovej metódy na systém modifikovaných Karush-Kuhn-Tuckerových podmienok optimality.

Lagrangeova funkcia pre úlohu (3.1) má tvar

$$L(x, \mu) = f_0(x) + \mu^T f(x),$$

kde funkcia $f(x) = (f_1(x), \dots, f_m(x))^T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ predstavuje ohraničenia a $\mu \geq 0_m$ je duálna premenná, ktorá prislúcha ohraničeniam $f(x) \leq 0$.

Vďaka platnosti Slaterovej podmienky pre úlohu (3.1) existuje primárno-duálna optimálna dvojica $(x^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^m$, ktorá spĺňa Karush-Kuhn-Tuckerove (KKT) podmienky, ktoré sú nutnými a postačujúcimi podmienkami optimality

$$\nabla f_0(x^*) + Df(x^*)^T \mu^* = 0 \quad (3.2)$$

$$\text{diag}(\mu^*) f(x^*) = 0 \quad (3.3)$$

$$f(x^*) \leq 0, \quad \mu^* \geq 0, \quad (3.4)$$

kde $Df(x)$ označuje Jakobiho maticu, t.j.

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

a $\text{diag}(\mu^*)$ je diagonálna matica so zložkami vektora μ^* na diagonále.

Myšlienkou primárno-duálnych metód vnútorného bodu je riešenie tohto systému Newtonovou metódou. Presnejšie, táto metóda sa aplikuje na systém nelineárnych rovníc

$$r_t(x, \mu) = \begin{pmatrix} \nabla f_0(x) + Df(x)^T \mu \\ -\text{diag}(\mu)f(x) - \frac{1}{t}\mathbf{1} \end{pmatrix} = 0. \quad (3.5)$$

Tento systém vznikol z prvých dvoch KKT podmienok, kde podmienka komplementarity (3.3) je upravená o člen $\frac{1}{t}\mathbf{1}$, pričom $t > 0$ je kladný parameter. Ak premenné x, μ spĺňajú $r_t(x, \mu) = 0$ a $f(x) < 0$, $\mu > 0$, potom $x = x^*(t)$, $\mu = \mu^*(t)$ sú body tzv. centrálnej trajektórie (x, μ sú vtedy prípustné s duálnou medzerou m/t). Všimnime si, že pre $t \rightarrow \infty$ sa systém rovníc (3.5) zredukuje na podmienky optimality pre pôvodnú úlohu (3.1). Preto je základnou myšlienkou primárno-duálnych metód používať Newtonove smery získané riešením systému (3.5) pre rastúcu postupnosť parametra t a pritom udržiavať kladnosť μ resp. zápornosť $f(x)$.

Majme bod $w = (x, \mu)$, ktorý spĺňa $f(x) < 0$, $\mu > 0$. Newtonova metóda pre (3.5) nájde korene linearizovaného systému

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

kde $Dr_t(w)$ predstavuje Jakobiho maticu a $\Delta w = (\Delta x, \Delta \mu)$ je Newtonov smer. Preto je potrebné riešiť systém lineárnych rovníc (nazývaný aj Newtonov lineárny systém) $Dr_t(w)\Delta w = -r_t(w)$. Tento systém, po rozpísaní do zložiek pre Δx a $\Delta \mu$, možno formálne zapísať ako

$$\left(\begin{array}{cc|c} \nabla^2 f_0(x) + \sum_{i=1}^m \mu_i \nabla^2 f_i(x) & Df(x)^T & -\nabla f_0(x) - Df(x)^T \mu \\ -\text{diag}(\mu)Df(x) & -\text{diag}(f(x)) & \text{diag}(\mu)f(x) + \frac{1}{t} \end{array} \right), \quad (3.6)$$

kde na ľavej strane pred zvislou čiarou je matica systému a na pravej strane za zvislou čiarou je pravá strana systému rovníc.

Pod primárno-duálnym smerom $\Delta w_{pd} = (\Delta x_{pd}, \Delta \mu_{pd})$ pre úlohu (3.1) rozumieme riešenie Newtonového systému lineárnych rovníc (3.6).

Poznámka 3.2.1. Metódy vnútorného bodu sú metódami druhého rádu a potrebujú riešiť systém lineárnych rovníc. Táto operácia má vo všeobecnosti výpočtovú zložitosť

$O(n^3)$, kde n je rozmer systému. V časti o implementácii primárno-duálnej metódy vnútorného bodu si povieme, akými spôsobmi možno k riešeniu Newtonovho systému pristupovať. Táto problematika je v prípade metód vnútorného bodu veľmi dôležitá, pretože výpočet tohto systému je výpočtovo (aj časovo) najnáročnejšia operácia v algoritmoch. Tieto prístupy môžu znamenať rozdiel medzi úspešným a neúspešným použitím metódy a dokážu metódu výrazne urýchliť.

3.2.2 Primárno-duálny algoritmus metódy vnútorného bodu

Štruktúra algoritmov metód vnútorného bodu je výrazne zložitejšia ako v prípade proximálnych metód. V nasledujúcej časti uvedieme základné prvky primárno-duálneho algoritmu.

Mierou optimality η je tzv. duálna medzera, ktorá je pre každé (x, μ) , ktoré spĺňajú $f(x) < 0$ a $\mu > 0$ definovaná ako

$$\eta(x, \mu) = -f(x)^T \mu.$$

Algoritmus primárno-duálnej metódy sa zastaví, keď je hodnota duálnej medzery menšia ako vopred zvolená tolerancia $\epsilon > 0$. Typicky je táto tolerancia malá, pretože primárno-duálne metódy dosahujú často lepšiu ako lineárnu rýchlosť konverencie [14].

Riadiaci parameter t je funkciou aktuálnej duálnej medzery η , pričom ho volíme ako $t = \nu m / \eta(x, \mu)$, kde konštanta $\nu > 1$ je zvyčajne z intervalu $[2, 10]$. Ak sú premenné x a μ centrálné, t.j. spĺňajú $r_t(x, \mu) = 0$ a $f(x) < 0$, $\mu > 0$, tak je duálna medzera rovná $\eta(x, \mu) = m/t$, a preto v tomto prípade hodnotu t zväčšujeme ν -násobne (t.j. $t := \nu t$) [14].

Ďalšou dôležitou súčasťou algoritmu primárno-duálnej metódy je určovanie dĺžky kroku. Veľkosť kroku z Newtonovej metódy skracujeme tak, aby boli splnené nerovnosti $\mu > 0$ a $f(x) < 0$. Za týmto účelom sa používa nasledujúci postup:

Označme súčasný bod ako w a bod novej iterácie ako w^+ , potom $w^+ = w + s\Delta w_{pd}$, kde s predstavuje dĺžku kroku. Najprv vypočítame s^{max} , maximálnu kladnú dĺžku kroku, ktorá spĺňa $\mu^+ = \mu + s\Delta\mu \geq 0$ ako

$$\begin{aligned} s^{max} &= \sup\{s \in [0, 1] \mid \mu + s\Delta\mu \geq 0\} \\ &= \min\{1, \min\{-\mu_i / \Delta\mu_i \mid \Delta\mu_i < 0\}\}. \end{aligned}$$

Následne položíme $s = 0.99s^{max}$ a s násobíme parametrom $\beta \in (0, 1)$ (skrácujeme dĺžku kroku), až do splnenia podmienok

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

Parameter α má zvyčajne hodnotu z intervalu $[0.01, 0.1]$ a parameter β má hodnotu z rozpätia $[0.3, 0.8]$ [14]. Podľa [14] sa príslušná hodnota s dá nájsť konečným počtom krokov.

Poznámka 3.2.2. V literatúre existujú viaceré myšlienkovito príbuzné postupy určovania dĺžky kroku, pozri napr. [105].

Algoritmus primárno-duálnej metódy vnútorného bodu na riešenie úlohy (3.1) vyzerá nasledovne:

Algoritmus - PDMVB pre riešenie (3.1)

- Vstup:**
- Štartovací bod $x : f(x) < 0; \mu > 0$.
 - Parametre $\nu > 1, \alpha, \beta$; tolerančné konštanty $\epsilon > 0, \epsilon_f > 0$.
- Výpočet:** Opakuj kroky 1, 2, 3, 4, 5, až do splnenia $\|r_{t_1}\|_2 \leq \epsilon_f$ a $\eta \leq \epsilon$.
- 1) Nastav $t = \nu m / \eta(x, \mu)$.
 - 2) Vypočítaj primárno-duálny smer $\Delta w_{pd} = (\Delta x_{pd}, \Delta \mu_{pd})$.
 - 3) Urči dĺžku kroku $s > 0$:
 - 3a) Nastav $s = 0.99 \sup\{s \in [0, 1] | \mu + s\Delta\mu \geq 0\}$.
 - 3b) Následne $s = \beta s$, až do splnenia

$$f(x^+) < 0, \quad \|r_t(x^+, \mu^+)\|_2 \leq (1 - \alpha s) \|r_t(x, \mu)\|_2.$$
 - 4) Výpočet novej iterácie $w = w + s\Delta w_{pd}$.
 - 5) Výpočet duálnej medzery $\eta(x, \mu) = -f(x)^T \mu$.

3.3 Nová PDMVB s redukciovou dimenziou

V tejto podkapitole predstavíme novú primárno-duálnu metódu vnútorného bodu, ktorú sme vytvorili špeciálne na riešenie l_1 -regulovaných úloh. Ako napovedá názov metódy, v jej algoritme budeme rozumným spôsobom postupne redukovať dimenziu dát, a tým pádom aj dimenziu riešených Newtonových systémov, ktorých výpočet predstavuje výpočtovo (aj časovo) najnáročnejšiu časť algoritmov MVB. Táto nová metóda predstavuje kombináciu algoritmu primárno-duálnych metód vnútorného bodu s algoritmom proximálnych metód, preto sa dá považovať za akúsi hybridnú metódu.

PDMVB s redukciovou dimenziou sa používa na riešenie l_1 -regulovanej úlohy na voľný extrém tvaru

$$\min_{x,y} F(x,y) + \lambda \|x\|_1, \quad (3.7)$$

kde $x \in \mathbb{R}^n$, $y \in \mathbb{R}$ a funkcia $F : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ je konvexná dvakrát spojitou diferencovateľná funkcia a $\lambda > 0$ je regularizačný parameter. Pod túto všeobecnú úlohu na voľný extrém regulovanú l_1 -normou spadajú aj úlohy (1.2) a (1.7) (úloha (1.2) však neobsahuje premennú y).

Hlavná myšlienka PDMVB s redukciovou dimenziou spočíva vo fakte, že l_1 -regulované úlohy nám poskytujú riedke riešenia x^* , t.j. riešenia, ktoré obsahujú malý počet nenulových zložiek. Ak sa nám podarí úspešne identifikovať množinu týchto nenulových zložiek, má zmysel optimalizáciu sústrediť len na danú množinu.

Autori v [46] sa zaoberajú l_1 -regulovanou úlohou tvaru

$$\min_{x \in \mathbb{R}^n} h(x) + \lambda \|x\|_1, \quad (3.8)$$

kde $h : \mathbb{R}^n \rightarrow \mathbb{R}$ je diferencovateľná konvexná funkcia a $\lambda > 0$. Autori v [46] dokázali, že ich algoritmus hľadania pevného bodu aplikovaný na l_1 -regulovanú úlohu (3.8) za pomerne slabých predpokladov vie nájsť pravé znamienka optimálneho riešenia (a teda aj množinu nenulových zložiek) za konečný počet iterácií, typicky skôr ako metóda skonverguje. Navyše tento algoritmus hľadania pevného bodu aplikovaný na úlohu (3.8) nájde množinu zložiek, ktorá obsahuje množinu všetkých nenulových zložiek pravého riešenia, omnoho skôr než odhalí skutočnú množinu [94] (pri nižšom počte iterácií metóda nájde aj „falošné“ nenulové zložky). Metóda hľadania pevného bodu z [46]

a [94] aplikovaná na úlohu (3.8) sa dá zapísať ako tzv. Forward-Backward splitting algoritmus tvaru

$$x_{k+1} = (I + tT_1)^{-1}(I - tT_2)x_k, \quad (3.9)$$

kde operátor $T_1(x) = \lambda \partial \|x\|_1$ a operátor $T_2(x) = \nabla h(x)$. Z toho vidíme, že sa jedná o algoritmus ekvivalentný aplikácii PGM na riešenie úlohy (3.8) (viď časť 2.2.2. Interpretácie PGM - PGM ako metóda hľadania pevného bodu). Preto budeme v našom algoritme používať PGM metódu.

K vytvoreniu novej metódy nás inšpiroval článok [94], v ktorom autori používajú hybridnú metódu HIS (Hybrid Iterative Shrinkage) na riešenie úlohy l_1 -regulovanej logistickej regresie. Táto hybridná metóda pozostáva z dvoch fáz. Prvá fáza využíva proximálnu metódu (modifikovanú PGM s využitím homotopie), po jej použití sa dimenzia dát jeden krát zredukuje a nasleduje druhá fáza pozostávajúca z aplikácie metódy vnútorného bodu (autori použili MVB z článku [58]) na zredukované dáta. Naš prístup pri PDMVB s redukciou dimenzie je však veľmi odlišný - náš algoritmus neobsahuje fázy, priamo sme spojili dve metódy do jednej. Jadro našej metódy tvorí stále primárnoduálna metóda vnútorného bodu a proximálnu gradientnú metódu využívame výlučne na redukovanie dimenzie dát (resp. identifikovanie množiny nenulových zložiek) a to v každej iterácii, pokiaľ sú splnené určité podmienky.

Idea hybridných algoritmov nie je nová. Avšak existujúce algoritmy nie sú nášmu prístupu veľmi príbuzné. Najpríbuznejší je práve už spomínaný algoritmus v [94]. Ďalší príbuzný algoritmus možno nájsť v [104], kde sa takisto používa myšlienka redukcie dimenzie pri riešení l_1 -regulovanej úlohy najmenších štvorcov. Autori z [104] sa spoliehajú že v prvej fáze algoritmu nájdú veľmi presnú množinu nenulových indexov riešenia spolu s jeho pravými znamienkami a v druhej fáze následne pristupujú k l_1 norme už ako k lineárnej funkcii a na riešenie novej úlohy s nižšou dimenzou používajú metódu konjugovaných gradientov či kvázinewtonovskú metódu BFGS. Prvá fáza týchto algoritmov však môže trvať dlho, preto sme sa rozhodli zvoliť iný prístup.

3.3.1 Algoritmus PDMVB s redukciou dimenzie

Algoritmus PDMVB s redukciou dimenzie v sebe kombinuje algoritmus PDMVB s algoritmom PGM metódy. Zatiaľ čo aplikácia PGM na riešenie (3.7) je priamočiara, v prípade PDMVB tomu tak nie je a úlohu (3.7) je potrebné transformovať. Algoritmus PDMVB s redukciou dimenzie tak pracuje s dvomi ekvivalentnými formuláciami úlohy (3.7) naraz.

Metódy vnútorného bodu sú používané na riešenie diferencovateľných úloh konvexného programovania tvaru (3.1). Úloha (3.7) (resp. (1.2), (1.7)) je však nediferencovateľná kvôli prítomnosti l_1 -normy v účelovej funkcii. Preto je potrebné danú úlohu najprv vhodne transformovať. Vytvorením novej premennej $z \in \mathbb{R}^n$ a pridaním nových ohraničení $-z_i \leq x_i \leq z_i$, $i = 1, \dots, n$, možno úlohu (3.7) previesť na konvexný problém s diferencovateľnou účelovou funkciou a s lineárnymi ohraničeniami. Ekvivalentná formulácia úlohy (3.7) má potom tvar

$$\begin{aligned} \min_{x,y,z} \quad & F(x,y) + \lambda \mathbf{1}^T z \\ \text{s.t.} \quad & x_i - z_i \leq 0, \quad i = 1, \dots, n, \\ & -x_i - z_i \leq 0, \end{aligned} \tag{3.10}$$

kde $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $z \in \mathbb{R}^n$ a $\mathbf{1}$ označuje vektor samých jednotiek, t.j. $\mathbf{1} = (1, \dots, 1)^T$.

Aby sme získali zápis algoritmu PDMVB s redukciou dimenzie, ktorý je kompaktnější, zrozumiteľnejší a takisto lepšie porovnateľný s algoritmom pôvodnej PDMVB metódy, označme funkcie

$$\begin{aligned} f_0(x,y,z) &= F(x,y) + \lambda \mathbf{1}^T z, \\ f(x,y,z) &= (x_1 - z_1, \dots, x_n - z_n, -x_1 - z_1, \dots, -x_n - z_n)^T. \end{aligned} \tag{3.11}$$

Potom dostaneme úlohu tvaru

$$\begin{aligned} \min_{x,y,z} \quad & f_0(x,y,z) \\ \text{s.t.} \quad & f(x,y,z) \leq 0, \end{aligned} \tag{3.12}$$

kde $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $z \in \mathbb{R}^n$ a funkcia $f_0 : \mathbb{R}^{2n+1} \rightarrow \mathbb{R}$ a funkcia $f : \mathbb{R}^{2n+1} \rightarrow \mathbb{R}^{2n}$. Postupy a odvodenia uvedené v časti 3.1 a 3.2 analogicky platia aj v prípade úlohy (3.12) (máme akurát viac premenných a ohraničení je $2n$).

Na identifikovanie množiny nenulových zložiek riešenia používame PGM metódu aplikovanú na úlohu (3.7). Do algoritmu PDMVB s redukciou dimenzie sme zabudovali PGM s dĺžkou kroku určenou v každej iterácii. Dĺžku kroku t_p zmeňujeme, až kým je splnená nerovnosť

$$\begin{aligned} F(\tilde{x}, \tilde{y}) \leq & F(x, y) + \nabla_x F(x, y)^T (\tilde{x} - x) + \\ & + \nabla_y F(x, y)^T (\tilde{y} - y) + \frac{1}{2t_p} (\|\tilde{x} - x\|_2^2 + \|\tilde{y} - y\|_2^2). \end{aligned} \quad (3.13)$$

Jedná sa o štandardnú podmienku používanú pri určovaní dĺžky kroku v proximal-gradientných algoritmoch [82].

Ako podmienky, pri ktorých splnení umožníme metóde redukovať dimenziu, používame nasledujúce vzťahy

$$\frac{\|\tilde{x} - x\|}{\max(\|x\|, 1)} < \epsilon_{p1}, \quad (3.14)$$

$$\frac{\|\nabla_x F(x, y)\|_\infty}{\lambda} - 1 < \epsilon_{p2}, \quad (3.15)$$

kde (x, y) je bodom z PDMVB a $(\tilde{x}, \tilde{y})$ je bod, ktorý vznikne následnou aplikáciou PGM metódy. Podmienky (3.14), (3.15) sú inšpirované podmienkami z [94]. Uvažovali sme k týmto podmienkam pridať aj podmienku s duálnou medzerou η či relatívnou suboptimalitou, avšak podmienky (3.14), (3.15) sa v našich experimentoch ukázali ako postačujúce pri správnom nastavení parametrov ϵ_{p1} a ϵ_{p2} . Hodnoty ϵ_{p1} sú zvyčajne z intervalu $[10^{-3}, 10^{-5}]$ a hodnoty ϵ_{p2} sú zvyčajne z intervalu $[0.2, 0.01]$. Podmienka (3.14) vyžaduje, aby relatívna zmena v premennej x bola pri použití PGM metódy malá. Podmienka je založená na myšlienke: Ak PGM metóda stagnuje, potom je veľmi pravdepodobné, že sme identifikovali všetky pravé nenulové zložky riešenia. Druhá podmienka (3.15) vychádza z podmienok optimality úlohy (3.7). Poznamenajme, že dobrá voľba parametrov ϵ_{p1} a ϵ_{p2} je dôležitá. Ak sú hodnoty parametrov nastavené príliš malé, k redukcii dochádza príliš neskoro a zrýchlenie metódy môže byť malé oproti pôvodnej PDMVB. Naopak, ak sú tieto parametre príliš veľké môže sa stať, že vyradíme nejakú pravú nenulovú zložku a už nevypočítame úlohu do takej presnosti ako pôvodne. V našich experimentoch sme však tieto hodnoty museli meniť len pri jedinej sade úloh.

Ak sú podmienky (3.14) a (3.15) splnené, povolíme algoritmu redukciu dimenzie, t.j. ponecháme len premenné a dáta prislúchajúce nenulovým zložkám bodu $\tilde{x}$ pochádzajúceho z PGM metódy. Ponecháme indexy $index = \{i | \tilde{x}_i \neq 0\}$ a následne premenné redukuje spôsobom $x = x(index)$, $z = z(index)$. Týmto spôsobom postupne redukuje aj vektor pôvodných indexov $((1, \dots, n))$, z ktorého na konci algoritmu zrekonštruujeme riešenie pôvodnej úlohy (3.7) (pretože udáva pôvodné polohy nenulových indexov). Predchádzajúcim postupom sa redukujú nielen dáta úlohy (pri maticiach dát A , resp. C ponechávame len stĺpce prislúchajúce nenulovým zložkám), ale aj počet ohraničení úlohy a k nim prislúchajúcich duálnych premenných. Tento jav má pre chod algoritmov metód vnútorného bodu niekoľko veľkých výhod:

- Tým, že postupne redukuje dimenziu matice dát, redukuje aj dimenziu riešených Newtonových systémov, ktorých výpočet predstavuje výpočtovo a časovo najnáročnejšiu časť algoritmov MVB. Čím presnejšie určíme pravú množinu nenulových zložiek riešenia, tým rýchlejšie budú prebiehať výpočty oproti pôvodnej PDMVB.
- Keďže MVB pracujú s tzv. vnútornými bodmi, ohraničenia $f_j \leq 0$, ktoré sú v optime splnené ako rovnosť, sú v algoritmoch MVB vždy vedené ako tesné ostré nerovnosti. Ak sa počas výpočtu dostaneme s bodmi iterácii príliš blízko k hranici množiny prípustných riešení P , môže to negatívne ovplyvniť čas a počet iterácii, ktoré MVB potrebujú na dosiahnutie riešenia. Takisto nás takéto ohraničenia môžu „odtláčať“ od presného riešenia. Tým pádom, ak je počet týchto ohraničení veľký, nemusíme dosiahnuť až tak presné riešenie. Keďže umožníme algoritmu PDMVB s redukciou dimenzie takéto ohraničenia redukovať, PDMVB metóde umožníme sústrediť optimalizáciu na tie podstatné zložky.
- Výsledné riešenie l_1 -regulovaných úloh pochádzajúce z MVB má zvyčajne všetky zložky nenulové a vyžaduje následné zaokrúhľovanie, napríklad na základe podmienok optimality úlohy. Výsledné riešenie z PDMVB s redukciou dimenzie má riedku štruktúru, častokrát ďalšie zaokrúhľovanie riešenia nepotrebuje.

Na základe prvých dvoch potenciálnych výhod očakávame rýchlejšie dosiahnutie presného riešenia oproti pôvodnej PDMVB.

Algoritmus primárno-duálnej metódy vnútorného bodu s redukciou dimenzie na riešenie úlohy (3.7), ((3.10), (3.12)) vyzerať nasledovne:

Algoritmus - PDMVB s redukciou dimenzie pre riešenie (3.7)

Vstup:	- Štartovací bod $(x, y, z) : f(x, y, z) < 0; \mu > 0$. - Parametre $\nu > 1, \alpha, \beta_1, \beta_2$; dĺžka kroku $t_p = t_0$; - Tolerančné konštanty $\epsilon > 0, \epsilon_f > 0, \epsilon_{p1} > 0, \epsilon_{p2} > 0$.
Výpočet:	Opakuj kroky 1, 2, 3, ..., 7 až kým $\ \nabla f_0 + Df^T \mu\ _2 \leq \epsilon_f$ a $\eta \leq \epsilon$. 1) Nastav $t = 2\nu n / \eta(x, y, z, \mu)$. 2) Vypočítaj primárno-duálny smer $\Delta w_{pd} = (\Delta x, \Delta y, \Delta z, \Delta \mu)$. 3) Urči dĺžku kroku $s > 0$: 3a) Nastav $s = 0.99 \sup\{s \in [0, 1] \mu + s\Delta\mu \geq 0\}$. 3b) Následne $s = \beta_1 s$, až do splnenia $f(x^+, y^+, z^+) < 0, \quad \ r_t(x^+, y^+, z^+, \mu^+)\ _2 \leq (1 - \alpha s) \ r_t(x, y, z, \mu)\ _2.$ 4) Výpočet novej iterácie $w = (x, y, z, \mu) = w + s\Delta w_{pd}$. 5a) Opakuj kroky 5b), 5c), 5d), 5e) 5b) $\tilde{x} = \text{prox}_{t_p \lambda \ \cdot\ }(x - t_p \nabla_x F(x, y))$. 5c) $\tilde{y} = y - t_p \nabla_y F(x, y)$. 5d) Preruš ak platí (3.13). 5e) $t_p = \beta_2 t_p$. 6) Ak sú splnené (3.14), (3.15) redukuj dimenziu dát. 7) Výpočet duálnej medzery $\eta(x, y, z, \mu) = -f(x, y, z)^T \mu$.

Zo zápisu algoritmu možno pozorovať, že jadro metódy tvorí stále PDMVB z predošlej časti 3.2. Kroky 5a)-5d), 6 predstavujú zabudovanie redukcie dimenzie do algoritmu, kde pomocou algoritmu PGM s určenou dĺžkou kroku rozhodujeme, či možno v danej iterácii redukovať dimenziu dát a ponechať len indexy prislúchajúce nenulovým zložkám bodu $\tilde{x}$ z PGM metódy. Poznamenajme, že aj keď je PGM metóda spádová a takisto sa vzdialenosť jej priebežného riešenia od optimálneho každou iteráciou znižuje, jej priebežné riešenie využívame čisto len na identifikáciu pravých nenulových zložiek riešenia (t.j. redukciiu dimenzie). Týmto spôsobom máme zaručené, že nerovnosti $f(x, y, z) < 0$ a $\mu > 0$, s ktorými pracuje PDMVB ostanú neporušené a v platnosti, akurát vyhadzujeme tie, čo identifikujeme ako nepodstatné.

Kapitola 4

Implementácia metód riešenia

V nasledujúcej kapitole sa budeme venovať implementácii proximálnych metód a primárno-duálnej metódy vnútorného bodu priamo na riešenie úlohy l_1 -regulovaných najmenších štvorcov (1.2) a úlohy l_1 -regulovanej logistickej regresie (1.7). Uvedieme tu dôležité komponenty (dĺžky krokov, proximálne operátory, Newtonove systémy a pod.) pre chod týchto metód, a takisto budeme dbať na efektívne vylepšenia/prístupy pri implementácii týchto metód. Predovšetkým u primárno-duálnej metódy vnútorného bodu môžu tieto vylepšenia predstavovať rozdiel „medzi dňom a nocou“ - napr. môžu metódu niekoľkonásobne urýchliť alebo umožnia metóde riešiť veľkorozmerné úlohy, pri ktorých by pred úpravou nevedela vypočítať ani jednu iteráciu kvôli zaplneniu operačnej pamäte.

4.1 Odvodenie a notácia základných stavebných prvkov algoritmov

Väčšina metód uvedených v predchádzajúcej kapitole využíva vo svojom algoritme informáciu z gradientov funkcie f , resp. z (čiastkových) gradientov funkcií f_i (prírásková proximal-gradientná metóda, PIAG, MISO). Primárno-duálna metóda vnútorného bodu bude pre svoj chod navyše požadovať Hessovu maticu funkcie f .

Pre úlohu l_1 -regulovaných najmenších štvorcov (1.2) máme

$$f(x) = \frac{1}{2} \|Ax - b\|_2^2 = \frac{1}{2m} \sum_{i=1}^m m(a_i^T x - b_i)^2, \quad (4.1)$$

$$\nabla f(x) = A^T(Ax - b) = \frac{1}{m} \sum_{i=1}^m m(a_i^T x - b_i)a_i, \quad (4.2)$$

$$\nabla^2 f(x) = A^T A, \quad (4.3)$$

kde $a_i \in \mathbb{R}^n$ je i -ty riadok matice A .

V prípade úlohy l_1 -regulovanej logistickej regresie (1.7) sú tieto funkcie nasledovné

$$f(x, y) = \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-b_i(a_i^T x + y))), \quad (4.4)$$

$$\nabla f(x, y) = \begin{pmatrix} \nabla_x f(x, y) \\ \nabla_y f(x, y) \end{pmatrix} = -\frac{1}{m} \sum_{i=1}^m \varphi(-b_i(a_i^T x + y)) \begin{pmatrix} b_i a_i \\ b_i \end{pmatrix}, \quad (4.5)$$

$$\nabla^2 f(x, y) = \frac{1}{m} \sum_{i=1}^m \varphi'(-b_i(a_i^T x + y)) \begin{pmatrix} b_i^2 a_i a_i^T & b_i^2 a_i \\ b_i^2 a_i^T & b_i^2 \end{pmatrix}, \quad (4.6)$$

kde funkcia φ a jej derivácia majú tvar

$$\varphi(\alpha) = \frac{\exp(\alpha)}{1 + \exp(\alpha)} = 1 - \frac{1}{1 + \exp(\alpha)}, \quad (4.7)$$

$$\varphi'(\alpha) = \frac{\exp(\alpha)}{(1 + \exp(\alpha))^2} = \frac{1}{1 + \exp(\alpha)} - \frac{1}{(1 + \exp(\alpha))^2}. \quad (4.8)$$

Pre mnohé algoritmy je vhodnejšie pracovať s maticovým zápisom. Ak vytvoríme novú maticu dát $C \in \mathbb{R}^{m \times n}$, definujeme vektor $\phi \in \mathbb{R}^m$ a diagonálnu maticu $D_\phi \in \mathbb{R}^{m \times m}$ nasledujúcim spôsobom

$$C = \begin{pmatrix} b_1 a_1, & \dots, & b_m a_m \end{pmatrix}^T, \quad \phi := \phi(x, y) = \begin{pmatrix} \varphi(-b_1(a_1^T x + y)) \\ \vdots \\ \varphi(-b_m(a_m^T x + y)) \end{pmatrix}, \quad (4.9)$$

$$D_\phi := D_\phi(x, y) = \begin{pmatrix} \varphi'(-b_1(a_1^T x + y)) & & \\ & \ddots & \\ & & \varphi'(-b_m(a_m^T x + y)) \end{pmatrix}, \quad (4.10)$$

predchádzajúce vzťahy pre gradient a Hessovu maticu pre funkciu f z úlohy (1.7)

možno ekvivalentne maticovo zapísať ako

$$\nabla f(x, y) = \begin{pmatrix} \nabla_x f(x, y) \\ \nabla_y f(x, y) \end{pmatrix} = -\frac{1}{m} \begin{pmatrix} C^T \\ b^T \end{pmatrix} \phi, \quad (4.11)$$

$$\nabla^2 f(x, y) = \frac{1}{m} \begin{pmatrix} C^T D_\phi C & C^T D_\phi b \\ b^T D_\phi C & b^T D_\phi b \end{pmatrix} = \frac{1}{m} \begin{pmatrix} C^T \\ b^T \end{pmatrix} D_\phi \begin{pmatrix} C & b \end{pmatrix}. \quad (4.12)$$

4.2 Implementácia proximálnych metód

V tejto kapitole približujeme ako správne aplikovať proximálne metódy na riešenie skúmaných úloh. Uvádzame užitočné informácie o proximálnych operátoroch a spôsobe voľby dĺžky kroku.

Keďže sa v prípade úloh (1.2) a (1.7) jedná o úlohy na voľný extrém, kde minimalizujeme funkciu rozložiteľnú na diferencovateľnú konvexnú časť a nediferencovateľnú konvexnú časť $\lambda\|x\|_1$, môžeme na ich optimalizáciu použiť proximálne metódy prezentované v predošlej časti.

4.2.1 Proximálne operátory

Pre aplikovanie proximálnych metód na riešenie úloh (1.2) a (1.7) je okrem štruktúry algoritmov potrebné poznať aj proximálne operátory niektorých funkcií. Pri proximálnych metódach sme spomínali, že tieto metódy sú vhodné predovšetkým v prípadoch, keď je proximálny operátor výpočtovo nenáročný, najlepšie explicitne vyjadriteľný.

Základný prvok proximálnych metód pre úlohy (1.2), (1.7) tvorí proximálny operátor $prox_{tg}$ pre nediferencovateľnú časť $g(u) = \lambda\|u\|_1 = \lambda \sum_{i=1}^n |u_i|$. Ako prvé si treba uvedomiť, že funkcia g je plne separovateľná. Preto vďaka vlastnosti proximálneho operátora z Tvrdenia 2.1.1. stačí vyjadriť jednu jeho zložku, pretože platí

$$(prox_{t\lambda\|\cdot\|_1}(x))_i = prox_{t\lambda|\cdot|}(x_i).$$

Využitím definície proximálneho operátora (2.1), a na základe nutných a postačujúcich podmienok optimality pre subdiferencovateľné funkcie ¹ dostávame

$$0 \in \partial|u_i^*| + \frac{1}{t\lambda}(u_i^* - x_i), \quad (4.13)$$

¹Pozri Príloha A.2 - Veta A.2.1.

kde $u_i^* = \text{prox}_{t\lambda|\cdot|}(x_i)$ a $\partial|u_i|$ je subdiferenciál funkcie, t.j.

$$\partial|u_i| = \begin{cases} -1 & u_i < 0 \\ [-1, 1] & u_i = 0 \\ 1 & u_i > 0 \end{cases}.$$

Vyjadrením bodu u_i^* z (4.13) možno jednoducho odvodiť výsledný tvar proximálneho operátora pre funkciu $g(u) = \lambda\|u\|_1$

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

Pri následnej aplikácii v programe MATLAB je vhodnejší vektorový zápis proximálneho operátora

$$\text{prox}_{tg}(x) = \text{sgn}(x)\max[0, |x| - t\lambda] \quad (4.14)$$

alebo ekvivalentne

$$\text{prox}_{tg}(x) = (x - t\lambda)_+ - (-x - t\lambda)_+. \quad (4.15)$$

Tieto zápisy citelne urýchlia chod algoritmov v prostredí MATLAB. V našich experimentoch sa ukázal byť mierne lepší tvar (4.14).

Poznámka 4.2.1. V prípade úlohy l_1 -regulovanej logistickej regresie (1.7), premenná y nevstupuje do l_1 -normy. Keďže proximálny operátor $\text{prox}_0(x) = x$, tak na premennú y aplikujeme v prípade základnej PGM iba štandardný gradientný krok.

ADMM

V prípade ADMM metódy je okrem proximálneho operátora nediferencovateľnej časti g potrebný aj proximálny operátor prox_{tf} pre diferencovateľnú časť f . Pri úlohe (1.2) je funkcia $f = \frac{1}{2}\|Ax - b\|_2^2$. Z definície proximálneho operátora (2.1) a použitím nutných a postačujúcich podmienok optimality pre konvexné diferencovateľné funkcie dostaneme vzťah

$$A^T(Au^* - b) + \frac{1}{t}(u^* - x) = 0,$$

kde $u^* = \text{prox}_{tf}(x)$. Z predošlého vzťahu možno jednoducho odvodiť tvar proximálneho operátora

$$\text{prox}_{tf}(x) = (A^T A + \frac{1}{t}I)^{-1}(A^T b + \frac{1}{t}x). \quad (4.16)$$

Vidíme, že v tomto prípade potrebujeme na určenie proximálneho operátora riešiť systém lineárnych rovníc, čo nie je ideálne, keďže chceme, aby bol proximálny operátor výpočtovo nenáročný. K riešeniu tohto systému sa dá pristúpiť viacerými spôsobmi. Napríklad sa dá riešiť priamo alebo iteračnými metódami (so štartom v bode z predošlej iterácie) alebo nájsť a uložiť do pamäte rozklad matice $(tA^T A + I)$ (ak dĺžku kroku t ponecháme fixnú). Prípadne, ak pre rozmery platí $m < n$, oplatí sa použiť Sherman–Morrison–Woodbury formula² a pracovať s menšou maticou $tAA^T + I$. Príbuzným systémom rovníc a ich riešeniu sa podrobnejšie venujeme pri aplikácii metód vnútorného bodu. Mnohé myšlienky z danej časti sa dajú priamočiaro aplikovať aj pre lineárny systém v ADMM.

V tejto práci sme sa pri experimentoch sústredili na prípad $m \leq n$, pre ktorý sme nami naprogramované metódy prispôbili. V prípade metódy ADMM sme sa rozhodli pre kombináciu dvoch vyššie uvedených taktík. Keďže pre väčšinu našich úloh platí $m < n$ pracujeme s maticou $tAA^T + I$ menších rozmerov. A takisto sme sa rozhodli ponechať konštantný krok, čo nám umožňuje vykonať 1 choleského rozklad matice na začiatku algoritmu a vo zvyšnej časti algoritmu pracujeme už len s jednoduchšie riešiteľnými trojuholníkovými systémami. Dúfame, že kombináciou týchto prístupov dostatočne zjednodušíme počítanie proximálneho operátora.

Priamo z Sherman–Morrison–Woodburyho formule³ získame rovnosť

$$(A^T A + \frac{1}{t}I)^{-1} = tI - t^2 A^T (I + tAA^T)^{-1} A. \quad (4.17)$$

Z predošlých vzťahov (4.16) a (4.17) vidíme, že proximálny operátor možno vypočítať nasledujúcim spôsobom: Najprv vyriešime lineárny systém

$$(I + tAA^T)p = A(A^T b + \frac{1}{t}x), \quad (4.18)$$

kde p je pomocná premenná. V druhom kroku dopočítame proximálny operátor

$$\text{prox}_{tf}(x) = tA^T b + x - t^2 A^T p. \quad (4.19)$$

²Pozri Príloha A.4.

³Pozri Príloha A.4.

Po použití Choleského rozkladu na maticu $I + tAA^T$ namiesto systému (4.18) v našej implementácii ADMM metódy riešime dva trojuholníkové systémy

$$L\tilde{p} = A(A^T b + \frac{1}{t}x), \quad (4.20)$$

$$L^T p = \tilde{p}, \quad (4.21)$$

kde L je dolná trojuholníková matica, ktorá spĺňa $I + tAA^T = LL^T$.

V prípade úlohy (1.7) je situácia zložitejšia, proximálny operátor funkcie f nie je explicitne vyjadriteľný a má tvar

$$\text{prox}_{tf}(x, y) = \underset{u, v}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-c_i^T u - b_i v)) \right) + \frac{1}{2t} (\|u - x\|_2^2 + (v - y)^2),$$

kde c_i je riadok matice C definovanej v (4.9). Preto by bolo pri implementácii ADMM na úlohu (1.7) potrebné pri výpočte proximálneho operátora funkcie f v každej iterácii riešiť úlohu na voľný extrém s diferencovateľnou účelovou funkciou. Tento fakt však znamená, že proximálny operátor je výpočtovo náročný, čo ovplyvňuje efektívne fungovanie metódy. Aj z tohto dôvodu sme sa rozhodli ADMM metódu na riešenie úlohy (1.7) l_1 -regulovanej logistickej regresie neaplikovať.

Poznámka 4.2.2. Predchádzajúca podúloha sa dá riešiť ľubovoľnou metódou voľnej optimalizácie, napríklad Newtonovou metódou. Poznamenajme, že ako štartovací bod je vhodné použiť bod z predchádzajúcej iterácie. Pri použití Newtonovej metódy pri hľadaní proximálneho operátora prox_{tf} treba pri určovaní smeru optimalizácie riešiť Newtonov systém, a preto by sa dali na vylepšenie chodu metódy uplatniť mnohé postupy z časti o aplikácii metód vnútorného bodu.

Poznámka 4.2.3. Pre efektívne použitie ADMM metódy na riešenie úlohy (1.7) by bolo vhodnejšie použiť paralelizovanú podobu algoritmu ADMM z [82], [16]. Paralelizácii algoritmov sme sa však v tejto práci nevenovali. V denšnej dobe je to však téma aktívneho výskumu z dôvodu narastajúceho počtu výpočtových jadier systémov.

4.2.2 Voľba dĺžky kroku

Ďalšou dôležitou súčasťou proximálnych metód je spôsob voľby dĺžky kroku t . Dĺžku kroku možno voľiť ako konštantnú avšak takisto existujú schémy ako určovať približne optimálnu dĺžku kroku v každej iterácii.

Proximálne metódy PGM, APG a ich modifikácie

Z teórie vieme, že konvergencia proximal-gradientnej metódy (PGM) a zrýchlenej proximal-gradientnej metódy (APGM) je zaručená pri konštantnom kroku $t = \frac{1}{L}$, kde L je Lipschitzovská konštanta gradientu funkcie f . V skutočnosti je konvergencia PGM garantovaná pre voľby kroku t z intervalu $(0, \frac{2}{L})$ [26], [46].

Pre úlohu (1.2) je funkcia $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, z Lipschitzovskej vlastnosti gradientu $\nabla f(x)$ dostaneme vzťah

$$\|\nabla f(x) - \nabla f(y)\|_2 = \|A^T A(x - y)\|_2 \leq \sigma_{\max}(A^T A)\|x - y\|_2,$$

kde $\sigma_{\max}(A^T A)$ predstavuje najväčšie vlastné číslo matice $A^T A$. Z toho dôvodu sa v algoritmoch proximálnych metód PGM, APGM a ich modifikáciách použije pri riešení úlohy (1.2) konštantný krok

$$t = \frac{1}{\sigma_{\max}(A^T A)}. \quad (4.22)$$

Pre úlohu (1.7) pri odvodzovaní Lipschitzovskej konštanty gradientu použijeme Tvrdenie A.1.1.⁴ o dvakrát spojitosti diferencovateľnej funkcie s Lipschitzovsky spojitým gradientom. Pre zjednodušenie zavedieme premennú $z = (x, y)^T$ a novú maticu dát $\tilde{C} = (C, b)$. Potom podľa (4.12) platí nasledujúce

$$\|\nabla^2 f(z)\|_2 = \frac{1}{m}\|\tilde{C}^T D_\phi \tilde{C}\|_2 = \frac{1}{m}\sigma_{\max}(\tilde{C}^T D_\phi \tilde{C}), \quad (4.23)$$

kde $\sigma_{\max}(\tilde{C}^T D_\phi \tilde{C})$ označuje najväčšie vlastné číslo matice $\tilde{C}^T D_\phi \tilde{C}$. Nech vektor $v \in \mathbb{R}^n$ je k nemu prislúchajúci vlastný vektor, potom dostaneme

$$\sigma_{\max}(\tilde{C}^T D_\phi \tilde{C})v^T v = v^T \tilde{C}^T D_\phi \tilde{C} v \leq \frac{1}{4}v^T \tilde{C}^T \tilde{C} v \leq \frac{\sigma_{\max}(\tilde{C}^T \tilde{C})}{4}v^T v, \quad (4.24)$$

kde $\sigma_{\max}(\tilde{C}^T \tilde{C})$ označuje najväčšie vlastné číslo matice $\tilde{C}^T \tilde{C}$. V predposlednej nerovnosti sme využili fakt, že matica D_ϕ je diagonálna s diagonálnymi prvkami z intervalu $(0, 1/4]$. Spojením vzťahov (4.23) a (4.24) dostaneme, že pre úlohu (1.7) platí

$$\|\nabla^2 f(z)\|_2 \leq \frac{\sigma_{\max}(\tilde{C}^T \tilde{C})}{4m}. \quad (4.25)$$

V algoritmoch PGM, APGM a ich modifikáciách sme preto pri riešení úlohy (1.7) použili konštantnú dĺžku kroku

$$t = \frac{4m}{\sigma_{\max}(\tilde{C}^T \tilde{C})}. \quad (4.26)$$

⁴Pozri Príloha A.1.

V literatúre existuje viacero techník ako určovať približne optimálnu dĺžku kroku pre PGM (APGM) metódu. Schémy na určovanie približne optimálneho kroku v každej iterácii možno nájsť napr. v [5], [101], [82]. Väčšina týchto algoritmov požaduje vyčísľovanie hodnoty účelovej funkcie (a iných hodnôt) v každej iterácii, čo narušuje jednoduchú stavbu metódy, a preto sa to nemusí vždy oplatiť (čo ukážeme aj v experimentálnej časti). V praxi však častokrát hodnotu Lipschitzovskej konštanty L nepoznáme alebo je jej samotný výpočet náročný (pre veľkorozmerné dáta). Podľa [5], algoritmus na určovanie približne optimálneho kroku pre proximálne metódy vyzerá nasledovne:

Algoritmus - určovanie dĺžky kroku pre PGM

Vstup: - Štartovací bod x_k , predošlý krok t , parameter $\beta \in (0, 1)$.

Výpočet: 1) Opakuj kroky 2), 3), 4)

2) $x = \text{prox}_{tg}(x_k - t\nabla f(x_k))$.

3) Preruš ak $f(x) \leq f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2t}\|x - x_k\|_2^2$.

4) $t = \beta t$.

5) Výstup: $x_{k+1} = x$.

Algoritmus určovania dĺžky kroku pre APGM vyzerá identicky, akurát namiesto štartovacieho bodu x_k používa extrapolovanú hodnotu y_k . Pri určovaní dĺžky kroku predchádzajúcim spôsobom PGM (APGM) dosahuje podobný konvergenčný výsledok ako pri voľbe konštantnej dĺžky kroku. Metóda dosahuje rovnakú sublineárnu rýchlosť konverencie funkčných hodnôt rádu $O(1/k)$ (resp. $O(1/(k+1)^2)$) ako vo Vete 2.2.1. (resp. Vete 2.2.2.) [101], [62].

Vyššie uvedený algoritmus určovania dĺžky kroku používame taktiež pre spádové modifikácie APGM. V prípade našej vlastnej spádovej modifikácie (Spádová DPG2) sme aplikovali tento postup na zrýchlenú časť a zároveň na časť, keď sme nútení používať základnú metódu. Navyše pre tieto časti povolíme používať rozdielne dĺžky krokov (čo experimentálne prinieslo malý prínos).

ADMM

Metóda ADMM konverguje pre všetky hodnoty parametra t za pomerne slabých podmienok [16]. Avšak rýchlosť tejto konvergenzie vie byť veľmi pomalá. Hodnota parametra t má výrazný vplyv na rýchlosť konvergenzie metódy. Vhodná voľba parametra t pre metódu ADMM pri rôznych triedach úloh je stále témou aktívneho výskumu, pozri napr. [41], [42], [72]. Keďže sme sa pri našej implementácii ADMM pre riešenie úlohy (1.2) rozhodli použiť Choleského rozklad, je potrebné používať konštantný parameter t . V experimentálnej časti otestujeme niektoré možné hodnoty parametra t , budeme testovať voľby kroku uvedené v [41]. V [41] sú uvedené konštantné kroky pre úlohu príbuznú našej, avšak ich funkčnosť je podložená iba experimentálne. Podľa [41] sme sa pre úlohu (1.2) rozhodli skúmať päťicu krokov:

$$t_1 = \frac{1}{\sigma_{\max}(A^T A)}, \quad t_2 = \frac{1}{\sqrt{\lambda \sigma_{\max}(A^T A)}}, \quad t_3 = \frac{1}{\lambda}, \quad t_4 = \frac{1}{\sigma_{\min}(A^T A)}, \quad t_5 = \frac{1}{\sqrt{\lambda \sigma_{\min}(A^T A)}}, \quad (4.27)$$

kde $\sigma_{\max}(A^T A)$ predstavuje najväčšie vlastné číslo matice $A^T A$ a $\sigma_{\min}(A^T A)$ predstavuje najmenšie nenulové vlastné číslo matice $A^T A$.

Prírastkové PG metódy

V prípade prírastkových metód sa pri voľbe dĺžky kroku v praxi často uplatňujú heuristické prístupy. Napríklad, autori v [75] navrhujú najprv prejsť 5% dát a na nich určiť dĺžku kroku tvaru $t = 2^k t_0$, ktorá pri výpočte dá najmenšiu hodnotu účelovej funkcie. Následne túto dĺžku kroku použijú pri aplikovaní metódy na celé dáta. Častokrát sa taktiež testujú kroky tvaru $t = \frac{1}{10^k}$ a vyberie sa najlepší z nich.

Autori SAG metódy, ktorej proximálnu verziu použijeme v experimentálnej časti práce, v článku [92] dokázali konvergenciu metódy pre voľbu konštantnej dĺžky kroku $t = \frac{1}{16L}$, kde L je spoločná Lipschitzovská konštanta gradientov ∇f_i , t.j. gradientov čiastkových funkcií f_i , ktorých priemer je prítomný v účelovej funkcii.

Pre úlohu (1.2) podľa (4.2) platí

$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 = \|m a_i a_i^T (x - y)\|_2 \leq m \|a_i\|_2^2 \|x - y\|_2, \quad (4.28)$$

čiže pre každý čiastkový gradient ∇f_i je Lipschitzovská konštanta $L_i = m \|a_i\|_2^2$ a

spoločná Lipschitzovská konštanta je rovná

$$L = \max_i L_i = m \max_i \|a_i\|_2^2. \quad (4.29)$$

Stochastická verzia PIAG (PSAG) metódy pri riešení úlohy (1.2) používa konštantný krok tvaru

$$t = \frac{1}{16m \max_i \|a_i\|_2^2}. \quad (4.30)$$

V prípade úlohy (1.7) pre zjednodušenie zavedieme premennú $z = (x, y)^T$ a novú maticu dát $\tilde{C} = (C, b)$, pričom $\tilde{c}_i$ označuje jej i -ty riadok. Pre odvodenie Lipschitzovskej konštanty L_i gradientu funkcie f_i potrebujeme ohraničiť vzťah

$$\|\nabla f_i(z_1) - \nabla f_i(z_2)\|_2 = \|(\varphi(-\tilde{c}_i^T z_1) - \varphi(-\tilde{c}_i^T z_2))\tilde{c}_i\|_2 = \|\varphi'(\theta)\tilde{c}_i\tilde{c}_i^T(z_2 - z_1)\|_2,$$

kde funkcia φ je definovaná v (4.7) a pri poslednej rovnosti sme použili Lagrangeovu vetu o strednej hodnote. Následne využitím faktu, že funkcia φ' nadobúda hodnoty z intervalu $(0, 1/4]$, získame

$$\|\varphi'(\theta)\tilde{c}_i\tilde{c}_i^T(z_2 - z_1)\|_2 \leq \frac{1}{4}\|\tilde{c}_i\tilde{c}_i^T(z_2 - z_1)\|_2 \leq \frac{\|\tilde{c}_i\|_2^2}{4}\|z_2 - z_1\|_2.$$

Každý čiastkový gradient ∇f_i má Lipschitzovskú konštantu $L_i = \frac{\|c_i\|_2^2}{4}$ a spoločná Lipschitzovská konštanta je rovná

$$L = \max_i L_i = \max_i \frac{\|c_i\|_2^2}{4}. \quad (4.31)$$

Stochastická verzia PIAG (PSAG) pri riešení úlohy (1.7) používa konštantnú dĺžku kroku tvaru

$$t = \frac{1}{4 \max_i \|c_i\|_2^2}. \quad (4.32)$$

V [92] je pre SAG metódu taktiež uvedený funkčný algoritmus určovania dĺžky kroku (resp. Lipschitzovskej konštanty) v každej iterácii, ktorý je nezávislý od počtu funkcií (t.j. rozmeru m). Schéma tohto algoritmu vyzerá nasledovne:

Algoritmus - určovanie dĺžky kroku pre PSAG

Vstup: - Štartovací bod x_k , index i_k , predošlý odhad Lip. konšt. L .

Výpočet: 0) Ak platí $\|\nabla f_{i_k}(x_k)\|^2 > 10^{-8}$ pokračuj.
 1) Polož $L := 2^{-1/m} L$.
 2) Opakuj kroky 3), 4).
 3) Preruš ak $f_{i_k}(x_k - \frac{1}{L} \nabla f_{i_k}(x_k)) \leq f_{i_k}(x_k) - \frac{1}{2L} \|\nabla f_{i_k}(x_k)\|^2$.
 4) $L = 2L$.
 5) Výstup: nový odhad Lipsch. konšt. L .

Podmienkou v bode 0) sa autori snažia vyhnúť numerickej nestabilite spôsobenej porovnávaním veľmi malých čísel. Všimnime si taktiež, že odhad L sa zmenší na polovicu ak podmienku 3) ihneď splníme m iterácií za sebou.

4.3 Implementácia metód vnútorného bodu

Aplikácia metód vnútorného bodu na riešenie úlohy l_1 -regulovaných najmenších štvorcov (1.2) a úlohy l_1 -regulovanej logistickej regresie (1.7) nie je taká priamočiara ako v prípade proximálnych metód. Metódy vnútorného bodu sú používané na riešenie diferencovateľných úloh konvexného programovania tvaru (3.1). Úlohy (1.2) a (1.7) sú však nediferencovateľné kvôli prítomnosti l_1 -normy v ich účelovej funkcii. Preto je potrebné dané úlohy najprv vhodne transformovať. Vytvorením novej premennej a nových ohraňení možno úlohy (1.2) a (1.7) previesť na konvexný problém s diferencovateľnou účelovou funkciou a s lineárnymi ohraňeniami, ktorý je riešiteľný metódami vnútorného bodu.

Keďže úlohy (1.2) a (1.7) majú podobnú štruktúru - zdieľajú zdroj svojej nediferencovateľnosti, ktorým je funkcia $g(x) = \lambda \|x\|_1$, môžu byť prevedené na diferencovateľnú úlohu rovnakými postupmi. Z toho dôvodu uvažujme všeobecnú úlohu na voľný extrém regulovanú l_1 -normou tvaru

$$\min_{x,y} f(x,y) + \lambda \|x\|_1, \quad (4.33)$$

kde $x \in \mathbb{R}^n$, $y \in \mathbb{R}$ a funkcia $f : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ je konvexná diferencovateľná funkcia. Pod túto všeobecnú úlohu spadajú aj nami riešené úlohy (1.2) a (1.7) (úloha (1.2) však

neobsahuje premennú y). Budeme uvažovať dva rôzne spôsoby transformácie úlohy (4.33) na diferencovateľnú úlohu.

4.3.1 Prvý spôsob transformácie l_1 -regulovanej úlohy

Úlohu (4.33) možno transformovať na ekvivalentnú diferencovateľnú formuláciu vytvorením novej premennej $z \in \mathbb{R}^n$ a pridaním nových ohraničení $-z_i \leq x_i \leq z_i$, $i = 1, \dots, n$. Ekvivalentná formulácia má potom tvar

$$\begin{aligned} \min_{x,y,z} \quad & f(x,y) + \lambda \mathbf{1}^T z \\ \text{s.t.} \quad & x - z \leq 0, \\ & -x - z \leq 0, \end{aligned} \tag{4.34}$$

kde $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $z \in \mathbb{R}^n$ a $\mathbf{1}$ označuje vektor samých jednotiek, t.j. $\mathbf{1} = (1, \dots, 1)^T$.

4.3.2 Druhý spôsob transformácie l_1 -regulovanej úlohy

Úlohu (4.33) taktiež možno previesť na ekvivalentnú diferencovateľnú formuláciu rozdelením premennej x na jej kladnú a zápornú časť. Nahradíme premennú x pomocou vzťahu $x = x^+ - x^-$, pričom $x^+ = \max(0, x)$ je kladná časť a $x^- = \max(-x, 0)$ je záporná časť. Lahko možno vidieť, že platí $\|x\|_1 = \|x^+ - x^-\|_1 = \mathbf{1}^T x^+ + \mathbf{1}^T x^-$. Danou transformáciou získame ekvivalentnú formuláciu tvaru

$$\begin{aligned} \min_{x^+, x^-, y} \quad & f(x^+ - x^-, y) + \lambda \mathbf{1}^T (x^+ + x^-), \\ \text{s.t.} \quad & x^+, x^- \geq 0 \end{aligned}$$

kde premenné $x^+ \in \mathbb{R}^n$, $x^- \in \mathbb{R}^n$ a $y \in \mathbb{R}$. V prípade úloh (1.2) a (1.7) sa táto formulácia dá ešte zjednodušiť. Ak zavedieme novú premennú $z = (x^+, x^-)^T \in \mathbb{R}^{2n}$ a novú maticu dát $\tilde{A} = [A, -A] \in \mathbb{R}^{m \times 2n}$ (resp. $\tilde{C} = [C, -C]$), získame zjednodušenú ekvivalentnú formuláciu

$$\begin{aligned} \min_{z,y} \quad & \tilde{f}(z, y) + \lambda \mathbf{1}^T z, \\ \text{s.t.} \quad & z \geq 0, \end{aligned} \tag{4.35}$$

kde $z \in \mathbb{R}^{2n}$ a $y \in \mathbb{R}$ a $\tilde{f} : \mathbb{R}^{2n+1} \rightarrow \mathbb{R}$ je funkcia používajúca novú maticu dát $\tilde{A}$ (resp. $\tilde{C}$).

Poznámka 4.3.1. Obe ekvivalentné formulácie (4.34) a (4.35) predstavujú úlohu konvexného programovania s konvexnou diferencovateľnou účelovou funkciou a s lineárnymi ohraničeniami. Pre úlohu l_1 -regulovaných najmenších štvorcov (1.2) majú tieto formulácie kvadratickú konvexnú účelovú funkciu, predstavujú teda úlohu kvadratického programovania. Formulácie (4.34) a (4.35) sú vhodné na aplikovanie metód vnútorného bodu.

Poznámka 4.3.2. V prípade oboch transformácií sa nám zavedením nových premenných (takmer) zdvojnásobil celkový počet premenných a taktiež nám pribudlo $2n$ ohraničení.

4.3.3 Primárno-duálny smer pre (4.34) a (4.35)

Základ primárno-duálnej metódy vnútorného bodu tvorí primárno-duálny smer, ktorý predstavuje smer optimalizácie primárnych a duálnych premenných. Získava sa aplikovaním Newtonovej metódy na systém modifikovaných Karush-Kuhn-Tuckerových podmienok optimality. Odvodíme preto KKT podmienky optimality pre obe diferencovateľné formulácie (4.34), (4.35), a taktiež odvodíme Newtonove systémy lineárnych rovníc, ktorých riešením je primárno-duálny smer optimalizácie. Celkovo teda získame 4 Newtonove systémy a to, 2 pre úlohu (1.2) a takisto 2 pre úlohu (1.7).

Primárno-duálny smer pre (4.34)

Lagrangeova funkcia pre úlohu (4.34) má tvar

$$L(x, y, z, \mu_1, \mu_2) = f(x, y) + \lambda \mathbf{1}^T z + \mu_1^T (x - z) + \mu_2^T (-x - z),$$

kde $x \in \mathbb{R}^n$, $y \in \mathbb{R}$, $z \in \mathbb{R}^n$ a $\mu_1 \geq 0$, $\mu_2 \geq 0$ sú duálne premenné, ktoré prislúchajú ohraničeniam $x - z \leq 0$, resp. $-x - z \leq 0$.

Karush-Kuhn-Tuckerove podmienky optimality pre úlohu (4.34) majú tvar

$$\nabla_x f(x, y) + \mu_1 - \mu_2 = 0, \quad \nabla_y f(x, y) = 0, \quad \lambda \mathbf{1} - \mu_1 - \mu_2 = 0,$$

$$\text{diag}(x - z)\mu_1 = 0, \quad \text{diag}(-x - z)\mu_2 = 0,$$

$$x - z \leq 0, \quad -x - z \leq 0, \quad \mu_1 \geq 0, \quad \mu_2 \geq 0.$$

Vynechaním ohraňení v tvare nerovností a perturbovaním podmienok komplementarity o člen $\frac{1}{t}\mathbf{1}$, kde $t > 0$ je kladný parameter, dostaneme nelineárny systém rovníc

$$r_t(x, y, z, \mu_1, \mu_2) = \begin{pmatrix} \nabla_x f(x, y) + \mu_1 - \mu_2 \\ \nabla_y f(x, y) \\ \lambda \mathbf{1} - \mu_1 - \mu_2 \\ -\text{diag}(x - z)\mu_1 - \frac{1}{t}\mathbf{1} \\ -\text{diag}(-x - z)\mu_2 - \frac{1}{t}\mathbf{1} \end{pmatrix} = 0.$$

Tento nelineárny systém následne riešime Newtonovou metódou, čiže je potrebné riešiť lineárny systém rovníc $Dr_t(w)\Delta w = -r_t(w)$, kde Dr_t predstavuje Jakobián r_t a riešenie $\Delta w = (\Delta x, \Delta y, \Delta z, \Delta \mu_1, \Delta \mu_2)$ je primárno-duálny smer optimalizácie. V premenných $\Delta x, \Delta y, \Delta z, \Delta \mu_1, \Delta \mu_2$ tento systém lineárnych rovníc vyzerá

$$\begin{pmatrix} \nabla_{xx}^2 f(x, y) & \nabla_{xy}^2 f(x, y) & 0 & I & -I & -\nabla_x f(x, y) - \mu_1 + \mu_2 \\ \nabla_{yx}^2 f(x, y) & \nabla_{yy}^2 f(x, y) & 0 & 0 & 0 & -\nabla_y f(x, y) \\ 0 & 0 & 0 & -I & -I & -\lambda \mathbf{1} + \mu_1 + \mu_2 \\ -M_1 & 0 & M_1 & Z - X & 0 & (X - Z)\mu_1 + \frac{1}{t}\mathbf{1} \\ M_2 & 0 & M_2 & 0 & X + Z & (-X - Z)\mu_2 + \frac{1}{t}\mathbf{1} \end{pmatrix},$$

kde $X = \text{diag}(x)$, $Z = \text{diag}(z)$, $M_1 = \text{diag}(\mu_1)$ a $M_2 = \text{diag}(\mu_2)$ sú diagonálne matice. Keďže by nebolo efektívne riešiť celý tento systém rovníc, systém čiastočne zeliminujeme na blokovo trojuholníkový tvar. Využijeme pritom fakt, že tento systém obsahuje množstvo diagonálnych matíc (s výnimkou Hessovej matice), ktorých inverzné matice sú ľahko počítateľné. Prenásobením štvrtej rovnice maticou $(Z - X)^{-1}$, piatej rovnice maticou $(X + Z)^{-1}$ a následným eliminovaním premenných $\Delta \mu_1$ a $\Delta \mu_2$ z prvej a tretej rovnice získame

$$\begin{pmatrix} \nabla_{xx}^2 f(x, y) - J_p & \nabla_{xy}^2 f(x, y) & J_m & 0 & 0 & -\nabla_x f(x, y) + s_1 - s_2 \\ \nabla_{yx}^2 f(x, y) & \nabla_{yy}^2 f(x, y) & 0 & 0 & 0 & -\nabla_y f(x, y) \\ J_m & 0 & -J_p & 0 & 0 & -\lambda \mathbf{1} - s_1 - s_2 \\ J_1^{-1} & 0 & -J_1^{-1} & I & 0 & -\mu_1 - s_1 \\ -J_2^{-1} & 0 & -J_2^{-1} & 0 & I & -\mu_2 - s_2 \end{pmatrix},$$

kde $J_1 = M_1^{-1}(X - Z)$, $J_2 = M_2^{-1}(-X - Z)$, $J_p = J_1^{-1} + J_2^{-1}$ a $J_m = J_1^{-1} - J_2^{-1}$ sú diagonálne matice a $s_1 = \frac{1}{t}(X - Z)^{-1}\mathbf{1}$, $s_2 = \frac{1}{t}(-X - Z)^{-1}\mathbf{1}$. Prenásobením tretej rov-

nice maticou $-J_p^{-1}$ a eliminovaním premennej Δz z prvej rovnice dostávame výsledný systém v blokovo trojuholníkovom tvare

$$\left(\begin{array}{ccccc|c} Q & \nabla_{xy}^2 f(x, y) & 0 & 0 & 0 & -\nabla_x f(x, y) + s_1 - s_2 - J_m J_p^{-1} v \\ \nabla_{yx}^2 f(x, y) & \nabla_{yy}^2 f(x, y) & 0 & 0 & 0 & -\nabla_y f(x, y) \\ -J_p^{-1} J_m & 0 & I & 0 & 0 & J_p^{-1} v \\ J_1^{-1} & 0 & -J_1^{-1} & I & 0 & -\mu_1 - s_1 \\ -J_2^{-1} & 0 & -J_2^{-1} & 0 & I & -\mu_2 - s_2 \end{array} \right),$$

pričom $v = \lambda \mathbf{1} + s_1 + s_2$ a $Q = \nabla_{xx}^2 f(x, y) - J_p + J_m J_p^{-1} J_m$ je symetrická kladne definitná matica.

Pre získanie primárno-duálneho smeru nám stačí vyriešiť prvé dve rovnice redukovaného systému (resp. jednu pre úlohu (1.2)), následne riešenie Δx použijeme na dopočítanie ďalších troch zložiek $\Delta z, \Delta \mu_1, \Delta \mu_2$. Výpočtovo najnáročnejšia časť riešenia je výpočet prvých zložiek $\Delta x, \Delta y$ primárno-duálneho smeru, preto si v ďalšej kapitole ukážeme, akým spôsobom možno takéto systémy efektívnejšie riešiť.

Úloha l_1 -regulovaných najmenších štvorcov (1.2) neobsahuje premennú y , preto treba riešiť systém lineárnych rovníc (využijeme vzťahy (4.2), (4.3))

$$(A^T A - J_p + J_m J_p^{-1} J_m) \Delta x = -A^T (Ax - b) + s_1 - s_2 - J_m J_p^{-1} v. \quad (4.36)$$

V prípade úlohy l_1 -regulovanej logistickej regresie (1.7), použitím vzťahov (4.11) a (4.12), získame systém rovníc tvaru

$$\begin{pmatrix} \frac{1}{m} C^T D_\phi C - J_p + J_m J_p^{-1} J_m & \frac{1}{m} C^T D_\phi b \\ \frac{1}{m} b^T D_\phi C & \frac{1}{m} b^T D_\phi b \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \frac{1}{m} C^T \phi + s_1 - s_2 - J_m J_p^{-1} v \\ \frac{1}{m} b^T \phi \end{pmatrix}, \quad (4.37)$$

ktorého riešením získame prvé zložky Newtonovho smeru.

Pri oboch úlohách ďalšie tri zložky Newtonovho smeru $\Delta z, \Delta \mu_1, \Delta \mu_2$ získame zo vzťahov

$$\begin{aligned} \Delta z &= J_p^{-1} (v + J_m \Delta x) \\ \Delta \mu_1 &= -\mu_1 - s_1 + J_1^{-1} (-\Delta x + \Delta z) \\ \Delta \mu_2 &= -\mu_2 - s_2 + J_2^{-1} (\Delta x + \Delta z), \end{aligned} \quad (4.38)$$

kde J_1, J_2, J_p a J_m sú diagonálne matice, a preto sú ich inverzie výpočtovo nenáročné.

Primárno-duálny smer pre (4.35)

Pre úlohu (4.35) je Lagrangeova funkcia tvaru

$$L(z, y, \mu) = \tilde{f}(z, y) + \lambda \mathbf{1}^T z - \mu^T z,$$

kde $z \in \mathbb{R}^{2n}$, $y \in \mathbb{R}$ a $\mu \geq 0$ je duálna premenná prislúchajúca ohraničeniu $z \geq 0$.

Karush-Kuhn-Tuckerove podmienky optimality pre úlohu (4.35) majú tvar

$$\nabla_z \tilde{f}(z, y) + \lambda \mathbf{1} - \mu = 0,$$

$$\nabla_y \tilde{f}(z, y) = 0,$$

$$-diag(z)\mu = 0,$$

$$z \geq 0, \mu \geq 0.$$

Vynechaním ohraničení v tvare nerovností a upravením podmienky komplementarity o člen $\frac{1}{t}\mathbf{1}$, kde $t > 0$ je kladný parameter, získame modifikované KKT podmienky. Primárno-duálny smer pre úlohu (4.35) je získaný aplikovaním Newtonovej metódy na systém nelineárnych rovníc

$$r_t(z, y, \mu) = \begin{pmatrix} \nabla_z \tilde{f}(z, y) + \lambda \mathbf{1} - \mu \\ \nabla_y \tilde{f}(z, y) \\ diag(z)\mu - \frac{1}{t}\mathbf{1} \end{pmatrix} = 0.$$

Preto je potrebné riešiť lineárny systém rovníc $Dr_t(w)\Delta w = -r_t(w)$, kde Dr_t predstavuje Jakobián r_t a riešenie $\Delta w = (\Delta z, \Delta y, \Delta \mu)$ je primárno-duálny smer. V premenných $\Delta z, \Delta y, \Delta \mu$ má tento systém lineárnych rovníc tvar

$$\begin{pmatrix} \nabla_{zz}^2 \tilde{f}(z, y) & \nabla_{zy}^2 \tilde{f}(z, y) & -I \\ \nabla_{yz}^2 \tilde{f}(z, y) & \nabla_{yy}^2 \tilde{f}(z, y) & 0 \\ diag(\mu) & 0 & diag(z) \end{pmatrix} \begin{pmatrix} -\nabla_z \tilde{f}(z, y) - \lambda \mathbf{1} + \mu \\ -\nabla_y \tilde{f}(z, y) \\ -diag(z)\mu + \frac{1}{t}\mathbf{1} \end{pmatrix}.$$

Kedže by nebolo efektívne riešiť celý tento systém rovníc, systém čiastočne zredukujeme na blokovo trojuholníkový tvar. Prenásobíme poslednú rovnicu maticou $diag(z)^{-1}$ a následne eliminujeme premennú $\Delta \mu$ z prvej rovnice, čím dostaneme výsledný systém

$$\begin{pmatrix} \nabla_{zz}^2 \tilde{f}(z, y) + J & \nabla_{zy}^2 \tilde{f}(z, y) & 0 \\ \nabla_{yz}^2 \tilde{f}(z, y) & \nabla_{yy}^2 \tilde{f}(z, y) & 0 \\ J & 0 & I \end{pmatrix} \begin{pmatrix} -\nabla_z \tilde{f}(z, y) - \lambda \mathbf{1} + s \\ -\nabla_y \tilde{f}(z, y) \\ -\mu + s \end{pmatrix},$$

kde $s = \frac{1}{t} \text{diag}(z)^{-1} \mathbf{1}$ a $J = \text{diag}(\mu) \text{diag}(z)^{-1}$ je diagonálna kladne definitná matica (a teda aj symetrická matica $\nabla_{zz}^2 \tilde{f}(z, y) + J$ je KD).

Pre získanie primárno-duálneho smeru optimalizácie potrebujeme vyriešiť prvé dve rovnice eliminovaného systému (resp. jednu pre úlohu (1.2)), následne riešenie Δz použijeme na dopočítanie ďalšej zložky $\Delta \mu$. Posledná zložka Newtonovho smeru $\Delta \mu$ sa dá jednoducho získať pre obe úlohy (1.2), (1.7) pomocou vzťahu

$$\Delta \mu = -J \Delta z - \mu + s. \quad (4.39)$$

Prvú zložku primárno-duálneho smeru Δz v prípade úlohy l_1 -regulovaných najmenších štvorcov (1.2) získame riešením lineárneho systému rovníc (úloha (1.2) neobsahuje premennú y , využijeme vzťahy (4.2), (4.3))

$$(\tilde{A}^T \tilde{A} + J) \Delta z = -\tilde{A}^T (\tilde{A} z - b) - \lambda \mathbf{1} + s, \quad (4.40)$$

kde $(\tilde{A}^T \tilde{A} + J) \in \mathbb{R}^{2n \times 2n}$ je symetrická kladne definitná matica a $\tilde{A} = [A, -A]$.

V prípade úlohy l_1 -regulovanej logistickej regresie (1.7), použitím vzťahov (4.11) a (4.12), získame pre prvé zložky primárno-duálneho smeru $\Delta z, \Delta y$ systém lineárnych rovníc tvaru

$$\begin{pmatrix} \frac{1}{m} \tilde{C}^T D_\phi \tilde{C} + J & \frac{1}{m} \tilde{C}^T D_\phi b \\ \frac{1}{m} b^T D_\phi \tilde{C} & \frac{1}{m} b^T D_\phi b \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta y \end{pmatrix} = \begin{pmatrix} \frac{1}{m} \tilde{C}^T \phi - \lambda \mathbf{1} + s \\ \frac{1}{m} b^T \phi \end{pmatrix}, \quad (4.41)$$

kde matica $\tilde{C} = [C, -C]$, pričom matica C je definovaná ako v (4.9).

Poznámka 4.3.3. Výpočet prvých zložiek $(\Delta x, \Delta y)$ v (4.36), (4.37), resp. $\Delta z, \Delta y$ v (4.40), (4.41)) primárno-duálneho smeru je výpočtovo najnáročnejšia časť pri hľadaní riešenia. Z toho dôvodu si v ďalšej kapitole ukážeme, akým spôsobom možno systémy tohto typu efektívnejšie riešiť.

4.3.4 Riešenie Newtonovych systémov lineárnych rovníc

Keďže metódy vnútorného bodu sú metódami druhého rádu, potrebujú vo svojom algoritme riešiť systém lineárnych rovníc, čo predstavuje ich hlavnú nevýhodu. Táto operácia má vo všeobecnosti výpočtovú zložitosť $O(n^3)$, kde n je rozmer systému. Primárno-duálny smer je riešením Newtonovho systému (3.6) získaného z modifikovaných KKT

podmienok optimality. Každá iterácia primárno-duálnej metódy vnútorného bodu je dominovaná cenou výpočtu primárno-duálneho smeru. Preto je potrebné pristupovať k riešeniu týchto systémov čo možno najefektívnejšie - treba využívať štruktúru dát či iteračné metódy.

Využitie štruktúry úlohy

Pri riešení Newtonových systémov lineárnych rovníc je vhodné využívať štruktúru dát, ako napríklad rozmery či riedkosť dátovej matice A . Na tento účel nám môžu poslúžiť algebraické triky uvedené v tejto časti.

Pre úlohu l_1 -regulovaných najmenších štvorcov (1.2) a úlohu l_1 -regulovanej logistickej regresie (1.7) počítame primárno-duálne smery optimalizácie (resp. prvé zložky týchto smerov) riešením systémov lineárnych rovníc (4.36), (4.40), resp. (4.37), (4.41). Tieto Newtonove systémy sú vhodné predovšetkým v prípade, ak pre rozmery matice A platí $n \ll m$. V prípadoch, keď pre maticu A platí $m \ll n$ alebo keď je matica A riedka (obsahuje málo nenulových členov) môže byť častokrát vhodnejšie riešiť iné ekvivalentné systémy.

Všimnime si, že Newtonove systémy (4.36) a (4.40) pre úlohu (1.2) sú tvaru

$$(A^T D_1 A + D_2) \Delta = g,$$

kde D_1 a D_2 sú diagonálne matice. Konkrétne, pre systém (4.36) sú matice $A \leftarrow A$, $D_1 \leftarrow I$ a $D_2 \leftarrow -J_p + J_m J_p^{-1} J_m$, a pre systém (4.40) sú matice $A \leftarrow \tilde{A} = [A, -A]$, $D_1 \leftarrow I$ a $D_2 \leftarrow J$.

Ak zavedieme pomocnú premennú $p = D_1 A \Delta$, potom možno predošlý systém ekvivalentne zapísať v tvare

$$\begin{pmatrix} D_2 & A^T \\ A & -D_1^{-1} \end{pmatrix} \begin{pmatrix} \Delta \\ p \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \quad (4.42)$$

Keď má matica A riedku štruktúru, matica $A^T A$ nemusí byť vždy dostatočne riedka (napríklad pri Truss Topology Design úlohe). V takom prípade je vhodnejšie namiesto Newtonových systémov (4.36), (4.40) riešiť systém v tvare (4.42). Navyše, pri tomto systéme nemusíme počítať a uskladňovať v pamäti maticu $A^T A$.

Systém tvaru (4.42) vieme ešte zredukovať. Ak si z prvej rovnice vyjadríme premennú $\Delta = D_2^{-1}(g - A^T p)$ a následne tento vzťah dosadíme do poslednej rovnosti, dostaneme lineárny systém

$$(AD_2^{-1}A^T + D_1^{-1})p = AD_2^{-1}g, \quad (4.43)$$

ktorý je vhodné použiť predovšetkým v prípade $m \ll n$. Newtonov smer (resp. prvé zložky) následne dopočítame afinnou transformáciou z riešenia predošlého systému ako $\Delta = D_2^{-1}(g - A^T p)$. Poznamenajme, že výsledný postup (zredukovaný tvar (4.43) a následné dopočítanie riešenia) je ekvivalentný použitiu Sherman–Morrison–Woodbury formuly⁵.

Analogický postup funguje aj pri úlohe (1.7). Systémy lineárnych rovníc (4.37) a (4.41) pre úlohu (1.7) majú tvar

$$\begin{pmatrix} C^T D_1 C + D_2 & C^T D_1 b \\ b^T D_1 C & b^T D_1 b \end{pmatrix} \begin{pmatrix} \Delta \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix},$$

kde D_1 a D_2 sú diagonálne matice. Konkrétne, pre systém (4.37) sú matice $C \leftarrow C$, $D_1 \leftarrow D_\phi/m$ a $D_2 \leftarrow -J_p + J_m J_p^{-1} J_m$, a pre systém (4.41) sú matice $C \leftarrow \tilde{C} = [C, -C]$, $D_1 \leftarrow D_\phi/m$ a $D_2 \leftarrow J$.

Zavedením pomocnej premennej $p = D_1 C \Delta$ pre prvú rovnicu možno predošlý systém ekvivalentne zapísať v tvare

$$\begin{pmatrix} D_2 & C^T & C^T D_1 b \\ C & -D_1^{-1} & 0 \\ b^T D_1 C & 0 & b^T D_1 b \end{pmatrix} \begin{pmatrix} \Delta \\ p \\ \Delta y \end{pmatrix} = \begin{pmatrix} g_1 \\ 0 \\ g_2 \end{pmatrix}. \quad (4.44)$$

Keď je matica C riedka, matica $C^T D_1 C$ nemusí mať dostatočne riedku štruktúru. V danom prípade je vhodnejšie namiesto Newtonových systémov (4.37), (4.41) riešiť systém v tvare (4.44). Ďalšia výhoda tohto postupu je, že pri tomto systéme nemusíme počítat a uskladňovať v pamäti maticu $C^T D_1 C$.

Lineárny systém (4.44) vieme ešte zredukovať. Použime substitúciu $p = D_1 C \Delta$ aj pre poslednú rovnicu. Ak si z prvej rovnice vyjadríme premennú $\Delta = D_2^{-1}(g_1 - C^T p -$

⁵Pozri Príloha A.4.

$C^T D_1 b \Delta y$) a následne tento vzťah dosadíme do druhej rovnosti, dostaneme lineárny systém

$$\begin{pmatrix} CD_2^{-1}C^T + D_1^{-1} & CD_2^{-1}C^T D_1 b \\ b^T & b^T D_1 b \end{pmatrix} \begin{pmatrix} p \\ \Delta y \end{pmatrix} = \begin{pmatrix} CD_2^{-1}g_1 \\ g_2 \end{pmatrix}, \quad (4.45)$$

ktorého riešenie použijeme na dopočítanie Newtonovho smeru (resp. prvých zložiek) pomocou $\Delta = D_2^{-1}(g_1 - C^T p - C^T D_1 b \Delta y)$. Lineárny systém tvaru (4.45) je vhodné použiť predovšetkým v prípade $m \ll n$.

Metódy riešenia systémov lineárnych rovníc - využitie iteračných metód

Výpočtovo najnáročnejšiu časť algoritmu primárno-duálnej metódy vnútorného bodu predstavuje počítanie primárno-duálneho smeru z Newtonovych systémov.

Prvou možnosťou ako Newtonove systémy riešiť sú priame metódy riešenia. Ich výhodou je, že dostaneme robustnú tzv. presnú (angl. exact) primárno-duálnu metódu využívajúcu presné Newtonove smery. Nevýhodou je výpočtová cena priamych metód, ktorá je vo všeobecnosti $O(n^3)$. Tento poznatok obmedzuje výkon metód vnútorného bodu používajúcich presné smery pre veľkorozmerné úlohy. Ak matica Newtonovho systému vykazuje riedkosť, možno systém riešiť priamymi metódami stavanými pre prácu s riedkými maticami.

V prípade metód vnútorného bodu však netreba riešiť Newtonove systémy presne. Za účelom vyhnutia sa drahému výpočtu Newtonovych smerov sa pri metódach vnútorného bodu využívajú iteračné metódy riešenia lineárnych systémov rovníc. Ako štartovací bod sa v nich zvyčajne volí Newtonov smer z prechádzajúcej iterácie. Takto získané približné riešenia Newtonovych systémov sa nazývajú nepresné (angl. inexact) Newtonove smery.

Použitie iteračných metód v algoritme môže zvýšiť počet hlavných iterácii metódy vnútorného bodu, čo je však často prevážené menšou výpočtovou náročnosťou. Iteračné metódy preto predstavujú akýsi kompromis medzi výpočtovou zložitou a rýchlosťou konverencie metódy. Použitie iteračných metód nám navyše umožňuje riešiť mnoho rozmernejšie úlohy vďaka ich menšej výpočtovej a pamäťovej náročnosti, čo si ukážeme aj v našich numerických experimentoch v tejto práci.

Kedže výkon iteračných metód závisí od spektrálnych vlastností matice systému,

v praxi sa používajú väčšinou prepodmienené iteračné metódy. Často sa používa prepodmienená metóda konjugovaných gradientov (PCG - Preconditioned Conjugate Gradient) či prepodmienená metóda minimálnych rezíduí (Minres). Pre viac detailov o metódach vnútorného bodu používajúcich nepresné smery pozri napr. [38], [56], [58], [45] a [21].

V našich experimentoch prevažne používame prepodmienenú metódu konjugovaných gradientov (PCG). Pri porovnaní s prepodmienenou metódou minimálnych rezíduí sme s PCG dosahovali konzistentne lepšie výsledky. PCG metóda vyžaduje, aby matica systému lineárnych rovníc bola symetrická kladne definitná. V našom prípade to spĺňajú systémy (4.36), (4.37), (4.40), (4.41), (4.43). Matice systémov (4.42), (4.44) sú symetrické no indefinitné, pri riešení týchto systémov sme použili prepodmienenú metódu minimálnych rezíduí. Matica systému (4.45) však nie je ani symetrická, na riešenie tohto systému iteračnou metódou sme použili prepodmienenú metódu bikonjugovaných gradientov (BICG), ktorá funguje pre ľubovoľnú $n \times n$ maticu.

Pri snahe dosiahnuť výpočtovo čo možno najefektívnejšie kódy, sme v Matlabe naprogramovali iteračnú metódu PCG. Vďaka lepšiemu prispôsobeniu našim úlohám sme s ňou dosahovali lepšie výsledky ako s PCG metódou predprogramovanou v prostredí Matlab. V prípade Minres a BICG sme používali ich predprogramované verzie z Matlabu. Algoritmus iteračnej metódy PCG pre riešenie systému $Hx = g$ vyzerá nasledovne:

Algoritmus - PCG (Preconditioned Conjugate Gradient)

- Vstup:** - Štartovací bod x_0 , symetrická kladne definitná matica H , vektor g , preconditioner P , konštanta presnosti ϵ_{PCG} .
- Výpočet:**
- 1) $r_0 = Hx_0 - g$, $Py_0 = r_0$, $p_0 = -y_0$, $k = 0$.
 - 2) Opakuj kroky 3), 4), ..., 9) až do splnenia $\frac{\|r_k\|_2}{\|g\|_2} \leq \epsilon_{PCG}$.
 - 3) $\alpha_k = \frac{r_k^T y_k}{p_k^T H p_k}$.
 - 4) $x_{k+1} = x_k + \alpha_k p_k$.
 - 5) $r_{k+1} = r_k + \alpha_k H p_k$.
 - 6) Vyrieš $Py_{k+1} = r_{k+1}$.
 - 7) $\beta_{k+1} = \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$.
 - 8) $p_{k+1} = -y_{k+1} + \beta_{k+1} p_k$.
 - 9) $k = k + 1$.

Poznámka 4.3.4. V praxi je počet PCG iterácií obmedzený maximálnym povoleným počtom iterácií a podávame najlepšie dosiahnuté riešenie do tohto limitu.

Všimnime si, že iteračná metóda PCG vo svojom algoritme maticu H používa len pri násobení vektorov. Z toho dôvodu je pre algoritmus PCG veľmi prospešné, ak má matica H zároveň riedku štruktúru. Fakt, že PCG (ako aj iné iteračné metódy) používa maticu H len pri násobení vektorov, má aj ďalšiu dôležitú implikáciu a výhodu. A to takú, že pri použití iteračných metód pri riešení Newtonovych systémov lineárnych rovníc nemusíme formovať (počítať) maticu H (napríklad pri systéme (4.37) nie je potrebné počítať maticu $C^T D_\phi C$).

Algoritmus PCG metódy a iných prepodmienенých iteračných metód používa tzv. preconditioner P , ktorý musí byť symetrický a kladne definitný. Preconditioner P by mal byť pomerne jednoduchý, ideálne taký, že poznáme jeho inverziu. V praxi sa preto často na prepodmieňovanie používajú diagonálne matice (tzv. Jacobiho prepodmieňovanie), či matice s diagonálnymi maticami v blokoch. V tejto práci používame preconditioner P práve týchto typov. Pre systém (4.36) napríklad používame preconditioner $P = kI - J_p + J_m J_p^{-1} J_m$ a pre (4.40) používame $P = k(I, -I; -I, I) + J$, kde k je kladný parameter. Pri prepodmieňovaní sme sa predovšetkým snažili odstrániť negatívny vplyv diagonálnych matic $-J_p + J_m J_p^{-1} J_m$ a J , ktoré pochádzajú priamo zo stavby primárno-

duálnych metód vnútorného bodu a sú častokrát veľmi zle podmienené. Z algoritmu PCG metódy si možno všimnúť, že pri iteračných metódach potrebujeme štartovací bod x_0 a takisto konštantu presnosti ϵ , do ktorej sa má daný lineárny systém riešiť. Ako štartovací bod je vhodné voliť primárno-duálny smer vypočítaný v predošlej iterácii. Konštantu presnosti ϵ v iteračných metódach určujeme adaptívnym pravidlom tvaru

$$\epsilon = \min(0.1, \frac{\xi\eta}{\|g\|_2}), \quad (4.46)$$

kde η je duálna medzera a parameter ξ nastavujeme podľa potrebnej presnosti. Parameter ξ má zvyčajne hodnotu z intervalu $[0.01, 0.001]$, avšak jeho vhodná hodnota je závislá od dát riešenej úlohy. Je potrebné ho dobre nastaviť - ak je jeho hodnota príliš malá, potom Newtonove systémy riešime príliš presne a robíme zbytočne veľa iterácii iteračnej metódy. Naopak, ak je jeho hodnota príliš veľká, nedosahujeme dostatočný pokles duálnej medzery v iteráciach PDMVB. Všimnime si, že použitím tohto pravidla, riešime každý Newtonov lineárny systém spočiatku s nízkou presnosťou (nie horšou ako 0.1) a postupne Newtonove systémy riešime s väčšou presnosťou ako sa duálna medzera v PDMVB metóde znižuje. Adaptívne pravidlo tohto tvaru sa pri nepresných MVB bežne používa [56], [58] a je príbuzné technikám používaných pri nepresných Newtonových metódach [81].

Časť II

Numerické experimenty a aplikácie

Kapitola 5

Metodika testovania a značenie metód

V nasledujúcich častiach práce otestujeme spomínané metódy riešenia úloh (1.2) a (1.7) v praxi. Na týchto experimentoch objasníme niektoré vlastnosti metód a porovnáme ich rýchlosti konverencie, pričom nás bude najviac zaujímať výpočtový čas. Experimenty sú uskutočnené v prostredí MATLAB na vlastných kódach. Všetky pokusy sú vykonané na PC zostave s procesorom Intel Core i9-9900K frekvencie 5GHz a s operačnou pamäťou 32GB.

V práci sme sa snažili čo možno najspravodlivejšie porovnať výkon rôznych metód. Keďže porovnáваме napríklad metódy prvého rádu s metódami druhého rádu, počet iterácií by nebola spravodlivá metrika, pretože každá metóda má inú výpočtovú cenu za iteráciu. Takisto rôzne typy metód zvyknú používať rôzne stopovacie kritériá, čo ešte viac komplikuje spravodlivé porovnanie. Rozhodli sme sa preto sledovať výpočtový čas a upraviť metódy tak, že sme im zabudovali jednotné stopovacie kritérium. Keďže metódy, ktoré potrebujú na výpočet viac iterácií, by boli týmto cudzím kritériom ovplyvnené viac ako napríklad PDMVB (ktoré zvyčajne problém vyriešia v desiatkach iterácií), rozhodli sme sa čas výpočtu stopovacieho kritéria nezarátať do výsledného času metód a jeho vplyv na výpočtový čas sme odstránili.

Pri numerických experimentoch v práci používame viacero generátorov dát / úloh podľa potreby experimentov. Pre úlohy, pri ktorých optimálne riešenie poznáme, ako

stopovacie kritérium v experimentoch používame relatívnu suboptimálnu

$$\frac{F - p^*}{p^*}, \quad (5.1)$$

kde F je hodnota účelovej funkcie v bode x pre (1.2) a p^* je optimálna hodnota účelovej funkcie.

Pre úlohy, ktorých optimálne riešenie nepoznáme (ktorých je väčšina) používame ako stopovacie kritérium duálnu medzeru predelenú hodnotou účelovej funkcie duálnej úlohy, ktorú získavame z čo najpresnejšie vyriešenej úlohy pomocou PDMVB. Používame teda vzťah

$$\frac{F - d}{d}, \quad (5.2)$$

kde F je hodnota účelovej funkcie v bode x pre (1.2), resp. v bode (x, y) pre (1.7) a d je presná hodnota duálnej účelovej funkcie z PDMVB. Tento vzťah predstavuje podľa slabej duality horné ohraničenie relatívnej suboptimality.

V experimentoch porovnávame výkon metód, t.j. dosiahnutie čo najpresnejšieho riešenia za čo najkratší čas. Pre lepší prehľad o výkonnosti metód vo väčšine experimentov tejto práce uvádzame výpočtový čas metód pre tri zvolené požadované presnosti

$$\epsilon_1 = 10^{-3}, \quad \epsilon_2 = 10^{-6}, \quad \epsilon_3 = 10^{-9}. \quad (5.3)$$

V jednotlivých metódach boli použité postupy a nastavenia parametrov metód, ktoré popisujeme v predošlej kapitole o implementácii metód. Pri mnohých metódach testujeme viaceré verzie. Napríklad pri metóde PGM používame metódu s konštantným krokom ale aj metódu, v ktorej určujeme dĺžku kroku t_k v každej iterácii. Pri primárno-duálnych metódach vnútorného bodu používame presné aj nepresné Newtonove smery a to pre všetky odvodené Newtonove systémy pre oba spôsoby transformácie úlohy (4.34) a (4.35). Preto máme celkovo 12 implementácii primárno-duálnych metód vnútorného bodu pre úlohu l_1 -regulovaných najmenších štvorcov (1.2) a takisto 12 pre úlohu l_1 -regulovanej logistickej regresie (1.7).

Z dôvodu veľkého počtu testovaných metód a prehľadnosti sme sa rozhodli vytvoriť skratky pre dané implementácie metód, ktoré budeme v experimentálnej časti práce opakovane používať. Zoznam a spôsob označenia používaných metód v experimentoch možno nájsť v nasledujúcej tabuľke.

ADMM	Alternating Direction Method of Multipliers
APG	zrýchlená proximal-gradientná metóda s konštantným krokom
APGls	zrýchlená proximal-gradientná metóda s určovaným krokom t_k
DPG1	spádová modifikácia APGM s konštantným krokom
DPG1ls	spádová modifikácia APGM s určovaným krokom t_k
DPG2	naša spádová modifikácia APGM s konštantným krokom
DPG2ls	naša spádová modifikácia APGM s určovaným krokom t_k
DPG3	APGM s reštartom a konštantným krokom
DPG3ls	APGM s reštartom a určovaným krokom t_k
MISOd	proximálna verzia deterministickej MISO metódy
MISOs	proximálna verzia stochastickej MISO metódy
PG	proximal-gradientná metóda s konštantným krokom
PGls	proximal-gradientná metóda s určovaným krokom t_k
PPGd	deterministická prírastková proximal-gradient metóda
PPGs	stochastická prírastková proximal-gradient metóda
PIAG	proximálna verzia deterministickej IAG metódy s konštantným krokom
PSAG	proximálna verzia stochastickej SAG metódy s konštantným krokom
PSAGls	proximálna verzia stochastickej SAG metódy s určovaným krokom t_k
PDxyz	primárno-duálne metódy vnútorného bodu, kde x je 1 pre prvý spôsob transformácie úlohy (4.34) x je 2 pre druhý spôsob transformácie úlohy (4.35) y je 1 pre Newtonov systém tvaru (4.36) ((4.37) pre úlohu (1.7)) alebo (4.40) ((4.41) úlohu (1.7)) y je 2 pre Newtonov systém tvaru (4.42) ((4.44) úlohu (1.7)) y je 3 pre Newtonov systém tvaru (4.43) ((4.45) úlohu (1.7)) z je <i>prázdne</i>, ak riešime Newtonov systém priamo z je i, ak riešime Newtonov systém iteračnými metódami
PD11ir	naša PDMVB s redukciou dimenzie vytvorená z PD11i

Všetky testované metódy sú našimi implementáciami metód v prostredí MATLAB. Prirodzene nás však môže napadnúť otázka, ako veľmi sa uvedené testované metódy, resp. ich modifikácie, líšia od známych metód, prípadne ako veľmi sme sa tvorivo podielali na ich implementácii. Metódy takto možno rozdeliť do štyroch skupín, pričom ich budeme v ďalšej časti práce podľa týchto skupín farebne odlišovať v tabuľkách experimentov.

V prípade pôvodných metód (čierna farba v tabuľkách) sme v Kapitole 4 o implementácii metód odvádzali hodnoty konštantných dĺžok krokov pre úlohy (1.2) a (1.7), a takisto v prípade metód APGLs a PSAGLs uvádzame štandardné algoritmy určovania dĺžky kroku v Kapitole 4. Metódy DPG1ls, DPG3ls, PGLs (modrá farba v tabuľkách) používajú jemne modifikovaný algoritmus určovania dĺžky kroku, o čom píšeme v Kapitole 6 v časti o proximálnych metódach. V prípade metódy ADMM (zelená farba v tabuľkách), o ktorej píšeme v Kapitole 2, sa jedná o náš prístup k implementácii metódy vzhľadom na vlastnosti nami generovaných dát, o čom píšeme a potrebné vzťahy odvádzame v Kapitole 4. Pri implementácii ADMM používame konštanté dĺžky krokov (4.27), vďaka čomu môžeme na začiatku algoritmu použiť Choleského rozklad na maticu menších rozmerov (použitie Sherman–Morrison–Woodburyho formuly) a následne efektívnejšie vyčíslovať hodnoty proximálneho operátora v algoritme metódy. Metódy PD xyz (zelená farba v tabuľkách) sú PDMVB, o ktorých píšeme a ich základný algoritmus uvádzame v Kapitole 3. Na aplikáciu týchto metód na l_1 -regulované úlohy je však tieto úlohy potrebné najprv transformovať, takisto je vhodné vzniknuté Newtonove systémy upraviť do lepšie riešiteľného tvaru a k samotnému riešeniu týchto Newtonovych systémov sa dá pristúpiť rôznymi spôsobmi, o týchto postupoch píšeme v Kapitole 4.3 o implementácii metód vnútorného bodu. Označenie xyz pri týchto metódach značí zvolený prístup pri aplikovaní PDMVB. Napríklad, metóda PD11 je PDMVB pre prvý spôsob transformácie (4.34) a používa Newtonov systém tvaru (4.36) a uvedený Newtonov systém rieši priamo. Metóda PD11i je tá istá metóda, avšak na riešenie Newtonovho systému využíva iteračné metódy. Metóda PD11i konkrétne používa prepodmienenú metódu konjugovaných gradientov (PCG), ktorú sme naprogramovali v prostredí Matlab a ktorej algoritmus uvádzame v časti 4.3. O nami vytvorenej modifikácii metódy APG, metóde DPG2 (červená farba v tabuľkách), píšeme a jej algoritmus

uvádzame v časti 2.2.4. Táto metóda je pôvodom z našej práce [62], no jej praktickú implementáciu sme značne vylepšili a implementácia pre riešenie úlohy (1.7) je úplne nová. Metódu DPG2ls (červená farba v tabuľkách) považujeme za novú, pretože využíva modifikovaný algoritmus určovania dĺžky kroku a navyše povoľuje používať iné dĺžky kroku pre PG a APG časti metódy. Metóda PD11ir (červená farba v tabuľkách) je nová PDMVB s redukciou dimenzie, ktorú sme navrhli a schému jej algoritmu uvádzame v Kapitole 3.3. Konkrétne metóda PD11ir vychádza priamo z úpravy algoritmu metódy PD11i, kde sme pomocou poznatkov z proximal-gradientných metód zabudovali postupnú redukciju dimenzie dát, s ktorými metóda pracuje.

Kapitola 6

Numerické experimenty

V súčasnej dobe vyžadujú aplikácie úloh l_1 -regulovaných najmenších štvorcov a l_1 -regulovanej logistickej regresie častokrát dáta veľkých rozmerov, ktoré majú navyše často riedku štruktúru. l_1 -regulované úlohy (1.2) a (1.7) majú veľký význam predovšetkým pri výbere podstatných charakteristík a proti tzv. over-fitu pre úlohy, kde je počet charakteristík n väčší ako počet dátových bodov/meraní m .

Naša základná úloha pre (1.2) aj pre (1.7) má preto počet charakteristík $n = 12000$ a počet dátových bodov $m = 2400$, pričom dátová matica $A \in \mathbb{R}^{2400 \times 12000}$ má riedku štruktúru s hustotou zaplnenia 5% ($de = 0.05$). Ako základné nastavenie regularizačného parametra λ sme zvolili hodnotu $\lambda = 0.1\lambda_{max}$, pri ktorej je počet nenulových zložiek riešenia pomerne blízky $m/2$. Pri numerických experimentoch následne testujeme vplyv zmeny rôznych parametrov tejto základnej úlohy na výkon metód riešenia.

Pre tieto úlohy sme sa snažili metódy prispôbiť a čo najefektívnejšie naprogramovať, aby sa šetrili výpočtové operácie a tým pádom aj výpočtový čas. Pri metódach boli použité postupy a nastavenia parametrov spomínané v kapitole o implementácii metód ako aj postupy z nasledujúcej podkapitoly.

6.1 Metódy riešenia

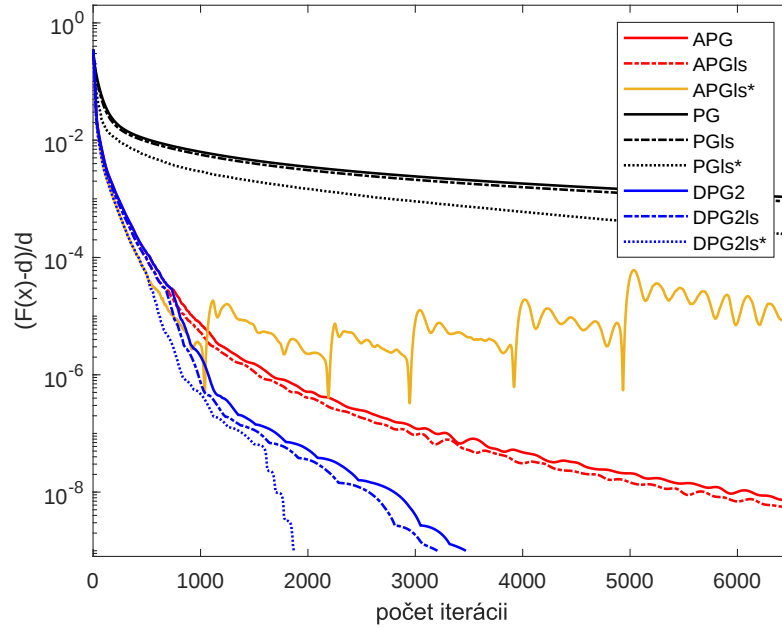
V nasledujúcej časti poukážeme na niektoré vlastnosti testovaných metód ako napríklad vývoj relatívnej suboptimality riešení pri použití metód, vplyv rôznych spôsobov voľby dĺžky kroku, vplyv predpodmienenia pri použití iteračných metód v PDMVB a objas-

níme fungovanie novej PDMVB s redukciou dimenzie. Takisto pri PDMVB uvádzame základné nastavenia parametrov metód.

6.1.1 Proximálne metódy

V kapitole o implementácii proximálnych metód sme odvodili hodnoty konštantných dĺžok kroku pre úlohy (1.2) a (1.7), a takisto sme uviedli bežne používaný algoritmus určovania dĺžky kroku pre proximal-gradientné metódy. Pomocou jednoduchšej modifikácie tohto algoritmu je však možné ešte vylepšiť výkon v praxi u väčšiny metód, ktoré ho používajú. Po úspešnom kroku metódy povolíme nájdenú dĺžku kroku zväčšiť o 10%. Touto úpravou už nemáme monotónny algoritmus určovania dĺžky kroku. Hlavná myšlienka tejto úpravy je tá, že Lipschitzovská konštanta L , ktorá sa vyskytuje v konštantných krokoch a ktorú sa snažíme odhadnúť algoritmom určovania dĺžky kroku je globálnou kvantitou. Avšak v neskorších fázach priebehu metód riešenia sa optimalizácia sústreďuje v okolí optimálneho riešenia. Je teda možné, že nejaká menšia lokálna kvantita (t.j. väčšia dĺžka kroku) môže fungovať v tomto okolí lepšie. K tejto myšlienke nás priviedli články [92], [75], podobnú myšlienku možno nájsť aj v článku o homotopickej proximal-gradientnej metóde [46].

Na Obr. 6.1 sa nachádza porovnanie výkonu proximálnych metód PG, APG, DPG2 s ich verziami, v ktorých je dĺžka kroku určovaná v každej iterácii. V prípade spádových metód PG, DPG1, DPG2, DPG3 spomínaná modifikácia algoritmu určovania dĺžky kroku poskytla signifikantný prínos pri našich experimentoch, čo možno pozorovať aj na Obr. 6.1 (viď PGls*, DPG2ls*). Všimnime si, že pre metódu APG však táto modifikácia na Obr. 6.1 nefunguje, aj keď pre väčšinu nami testovaných úloh fungovala. Dôvod je jednoduchý, na garantovanie konvergenzie metódy APG je potrebný monotónny algoritmus určovania dĺžky kroku, čo možno vidieť z dôkazov konvergenzie APG metódy napr. v [101], [62]. PG metóda takúto požiadavku nemá a spádové modifikácie DPG1, DPG2 a DPG3 obsahujú vo svojom algoritme podmienku nerastúcej účelovej funkcie. Z týchto dôvodov budeme v ďalšej časti práce pre všetky proximal-gradientné metódy okrem APG používať pri určovaní dĺžky kroku túto modifikáciu. Pri použití týchto metód sme mnohokrát kontrolovali, či daná modifikácia predstavuje pre metódy prínos z hľadiska redukcie výpočtového času, čo sa nám vždy potvrdilo.

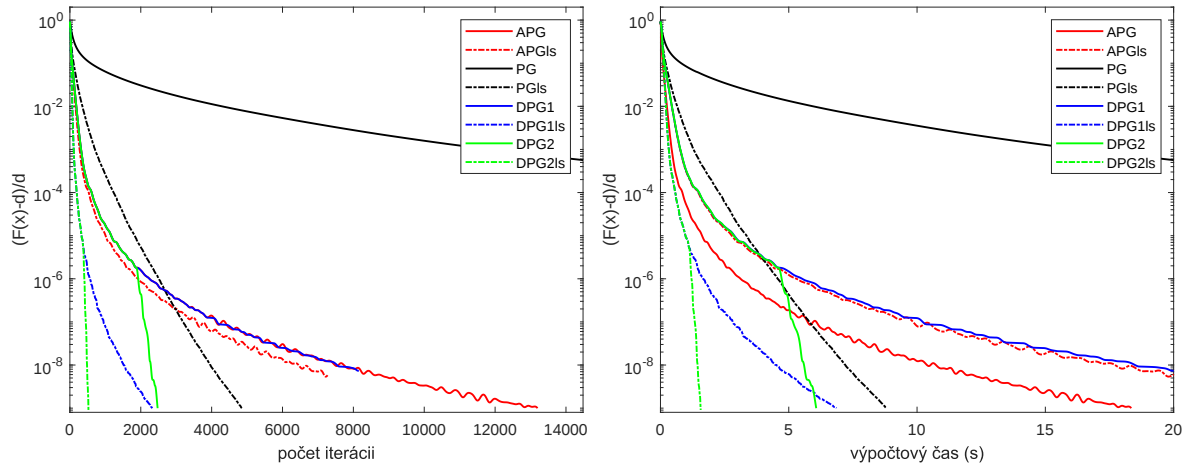


Obr. 6.1: Porovnanie výkonu proximal-gradientných metód PG, APG a DPG2 pre odlišné spôsoby určovania dĺžky kroku na náhodne generovanej úlohe (1.2).

Na Obr. 6.2 sa nachádza porovnanie výkonu proximal-gradientných metód pre náhodne generovanú úlohu logistickej regresie (1.7). Možno pozorovať, že všetky metódy s určenou dĺžkou kroku podávajú z hľadiska počtu iterácií lepší výkon ako ich verzie používajúce konštantnú dĺžku kroku (4.26). Identické pozorovanie platí z hľadiska výpočtového času pre všetky metódy okrem metódy APG. Tento jav je spôsobený aj tým, že pridaním algoritmu určovania dĺžky kroku sme narušili jednoduchú stavbu metódy a v iteráciách musí metóda navyše počítať ďalšie funkčné hodnoty. Na obrázku možno taktiež vidieť typické „skoky“ funkčných hodnôt pri použití APG metódy ako aj spádovosť ostatných metód. APG metóda a jej spádové modifikácie DPG1 a DPG2 podávajú identické riešenie až po prvý nárast funkčnej hodnoty pri APG metóde, tento jav je možné v práci bežne pozorovať, súvisí so stavbou algoritmov metód.

ADMM

Metóda ADMM konverguje pre všetky hodnoty parametra t , avšak hodnota tohto parametra má výrazný vplyv na rýchlosť konverencie metódy. Keďže sme sa pri našej implementácii ADMM pre riešenie úlohy (1.2) rozhodli použiť Choleského rozklad, je



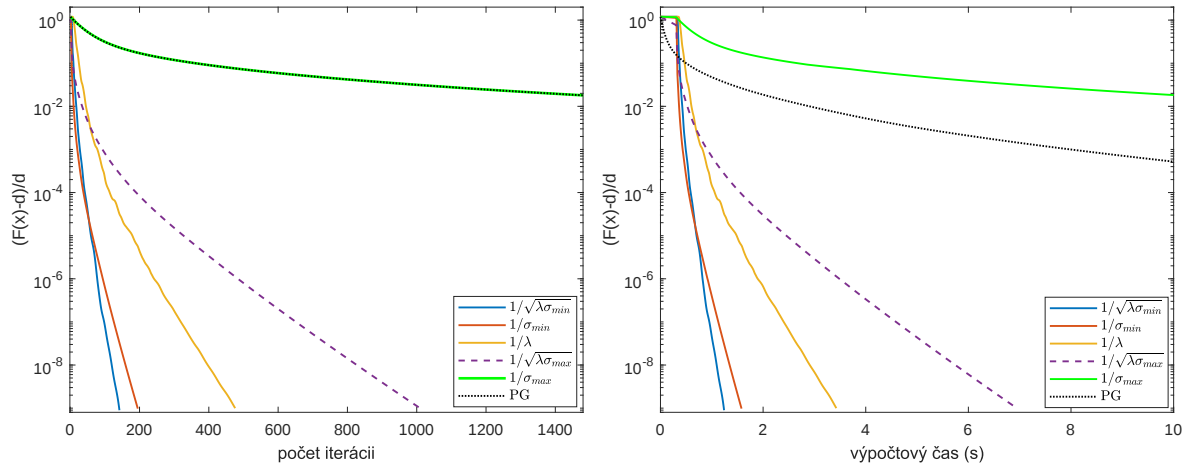
Obr. 6.2: Porovnanie výkonu proximal-gradientných metód PG, PGls, APG, APGls, DPG1, DPG1ls, DPG2, DPG2ls na náhodne generovanej úlohe (1.7).

potrebné používať konštantný parameter t . Na základe [41], kde sú uvedené konštantné kroky pre úlohu príbuznú úlohe (1.2), sme sa rozhodli testovať päťicu krokov (4.27). Výkon ADMM metódy pre rôzne hodnoty parametra t je zobrazený na Obr. 6.3. Na Obr. 6.3 je vyobrazená taktiež PG metóda s konštantnou dĺžkou kroku $t = 1/\sigma_{\max}(A^T A)$, kde $\sigma_{\max}(A^T A)$ je najväčšie vlastné číslo matice $A^T A$. Môžeme pozorovať, že PG metóda a ADMM metóda pri tej istej hodnote parametra t podávajú riešenie s rovnakou relatívnou suboptimalitou. Takisto si možno všimnúť čas výpočtu Choleského rozkladu na začiatku chodu ADMM metódy.

Poznamenajme, že výkon ADMM metódy pre rôzne hodnoty parametra t sa mení v závislosti od riešenej úlohy. Napriek našej snahe sme nenašli rozumný spôsob ako vopred určiť, ktorú hodnotu parametra t použiť. V priebehu našich experimentov boli najlepšími všetky testované hodnoty parametra okrem hodnoty $t = 1/\sigma_{\max}(A^T A)$. Metódu ADMM sme pri experimentoch na 50-tich úlohách používali tak, že na prvej úlohe testovacej sady sme určili najlepšiu voľbu parametra t , ktorú sme následne používali aj na ďalšie úlohy testovacej sady.

6.1.2 Prírastkové PG metódy

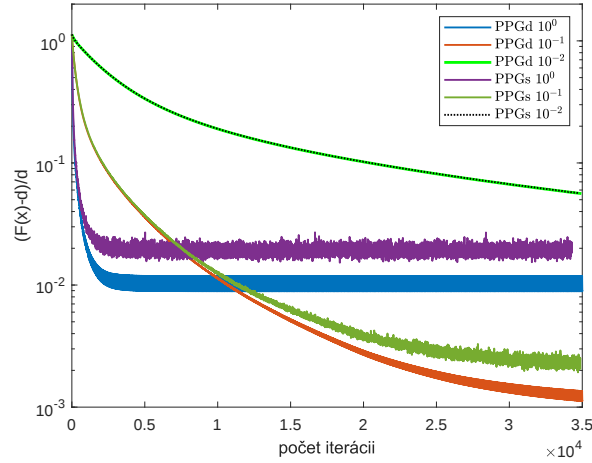
V prípade základných prírastkových metód sa pri voľbe dĺžky kroku častokrát testujú dĺžky krokov tvaru $t = \frac{1}{10^k}$, a následne sa vyberie najlepšia z nich. Na Obr 6.4 sa nachá-



Obr. 6.3: Porovnanie výkonu proximálnej metódy ADMM pre rôzne hodnoty parametra t a PG metódy na náhodne generovanej úlohe (1.2).

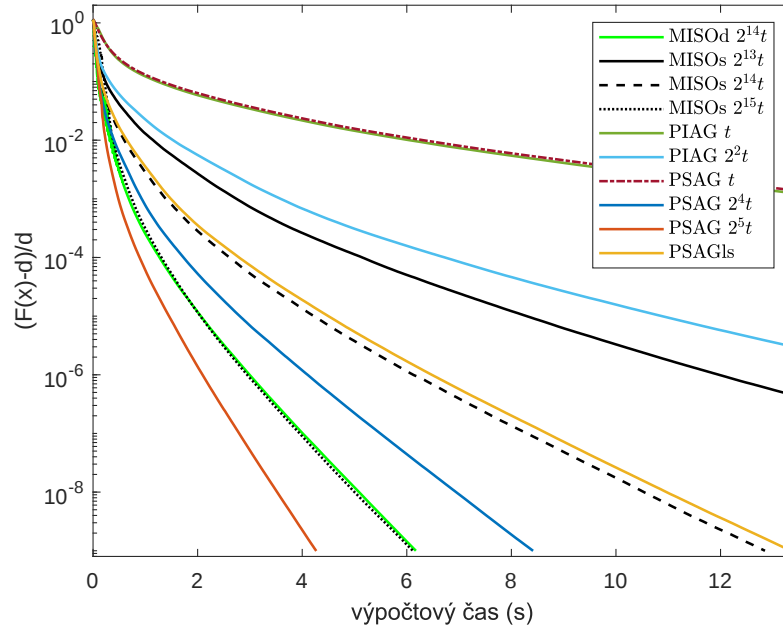
dza porovnanie výkonu deterministickej (PPGd) a stochastickej (PPGs) prírastkovej proximal-gradientnej metódy pre rôzne hodnoty dĺžky kroku. Môžeme pozorovať, že so zmenšovaním dĺžky kroku klesá variancia funkčných hodnôt. Taktiež možno pozorovať, že rýchlejšie dostaneme odhad riešenia pri použití väčšej dĺžky kroku. Z toho dôvodu sa zvyknú pri podobných metódach používať aj postupne klesajúce dĺžky krokov. Tieto základné metódy však v praxi dosahujú iba pomerne malú presnosť a majú pomalú rýchlosť konvergenzie.

Omnoho lepšie výsledky v praxi dosahujú modifikácie prírastkovej proximal-gradientnej metódy ako PIAG, PSAG, MISOd, MISOs. Porovnanie výkonu týchto prírastkových proximálnych metód pre rôzne voľby dĺžky kroku je na Obr. 6.5. Voľba správnej dĺžky kroku má na ich výkon veľký vplyv, so zväčšujúcou dĺžkou kroku sa zlepšuje výkon týchto metód. Autori SAG metódy, ktorej proximálnu verziu používame, v článku [92] dokázali konvergenciu metódy pre voľbu konštantnej dĺžky kroku $t = \frac{1}{16L}$, kde L je spoločná Lipschitzovská konštanta gradientov ∇f_i (viď Obr. 6.5: PIAG t a PSAG t). Autori taktiež uvádzajú, že SAG metóda v ich experimentoch fungovala aj pre voľbu dĺžky kroku $\frac{1}{L}$ (viď Obr. 6.5: PSAG $2^4 t$). V [92] je takisto uvedený algoritmus určovania dĺžky kroku v každej iterácii (viď Obr. 6.5: PSAGls), ktorého funkčnosť je podložená experimentálne. Tento algoritmus ako aj odvodenie dĺžky krokov pre úlohy (1.2), (1.7) možno nájsť v Kapitole o implementácii proximálnych metód.



Obr. 6.4: Porovnanie výkonu prírastkových proximálnych metód PPGd a PPGs pre rôzne hodnoty dĺžky kroku na náhodne generovanej úlohe (1.2).

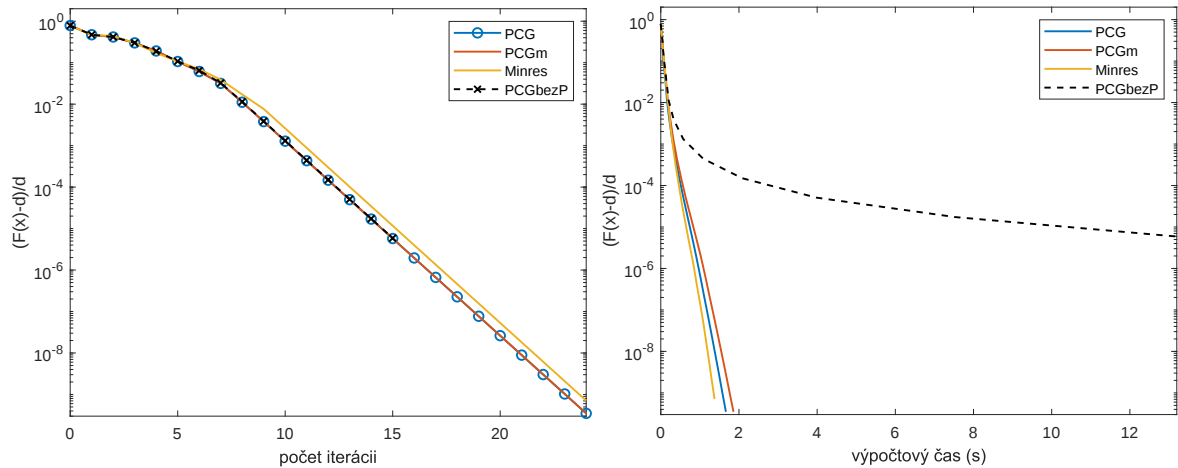
Pri metóde MISO autori v [75] navrhujú heuristický prístup voľby dĺžky kroku - navrhujú najprv prejsť 5% dát a na nich určiť dĺžku kroku tvaru $2^k t_0$, ktorá pri výpočte poskytne najmenšiu hodnotu účelovej funkcie. Následne túto dĺžku kroku použijú pri aplikovaní metódy na celé dáta. Z toho dôvodu na Obr. 6.5 uvádzame metódy s dĺžkami krokov tohto tvaru, pričom metódy MISOd $2^{14}t$, MISOs $2^{15}t$, PIAG 2^2t a PSAG 2^5t používajú najväčšie dĺžky krokov, pri ktorých boli metódy schopné konvergovať. Tento postup hľadania dĺžky kroku má okrem jeho veľkej časovej náročnosti aj ďalšiu nevýhodu. Viackrát sa nám pri stochastických metódach PSAG a MISOs podarilo nájsť takú dĺžku kroku, ktorá pri dobrej náhodnosti poskytla presné riešenie úlohy najrýchlejšie, avšak pri zlej náhodnosti metóda divergovala. Preto tento postup nepovažujeme za veľmi použiteľný. Z Obr. 6.5 si možno taktiež všimnúť, že stochastické verzie algoritmov nám umožňujú používať väčšie dĺžky krokov. Napriek tomu však metóda MISOs $2^{15}t$ dosahuje podobný výkon ako MISOd $2^{14}t$ pre menšiu voľbu parametra, tento jav sme pri experimentoch pozorovali viackrát. Naopak PSAG metóda podávala omnoho lepšie výsledky ako jej deterministická verzia PIAG.



Obr. 6.5: Porovnanie výkonu prírastkových proximálnych metód MISOd, MISOs, PIAG a PSAG pre rôzne hodnoty dĺžky kroku na náhodne generovanej úlohe (1.2).

6.1.3 PDMVB

Výpočtovo najnáročnejšou časťou algoritmov primárno-dálnych metód vnútorného bodu je počítanie primárno-duálnych smerov optimalizácie z Newtonových systémov. V prípade metód vnútorného bodu však nie je potrebné určovať tieto smery presne a na ich výpočet možno použiť iteračné metódy. Vo finálnych verziách väčšiny PDMVB metód využívajúcich iteračné metódy používame našu implementáciu prepodmienenej metódy konjugovaných gradientov. Porovnanie výkonu našej implementácie PCG metódy s PCG a Minres metódami predprogramovaných v MATLABe možno pozorovať na Obr. 6.6 a Obr. 6.7. Na náhodne generovanej úlohe (1.2) podáva naša implementácia PCG metódy podobný výkon ako PCG a Minres z Matlabu. Avšak v prípade TTD úlohy možno pozorovať medzi metódami veľké rozdiely a vďaka našej implementácii PCG sme ušetrili množstvo výpočtového času. Pri prepodmienených iter. metódach v práci používame prepodmieňovanie pomocou diagonálnych či blokovo diagonálnych matic. Efekt takéhoto prepodmieňovania možno taktiež vidieť na Obr. 6.6 a Obr. 6.7. Vidíme, že metóda PCG bez prepodmienenia podáva veľmi príbuzné riešenie ako prepodmienené metódy, avšak na jeho výpočet potrebuje omnoho viac výpočtového času.

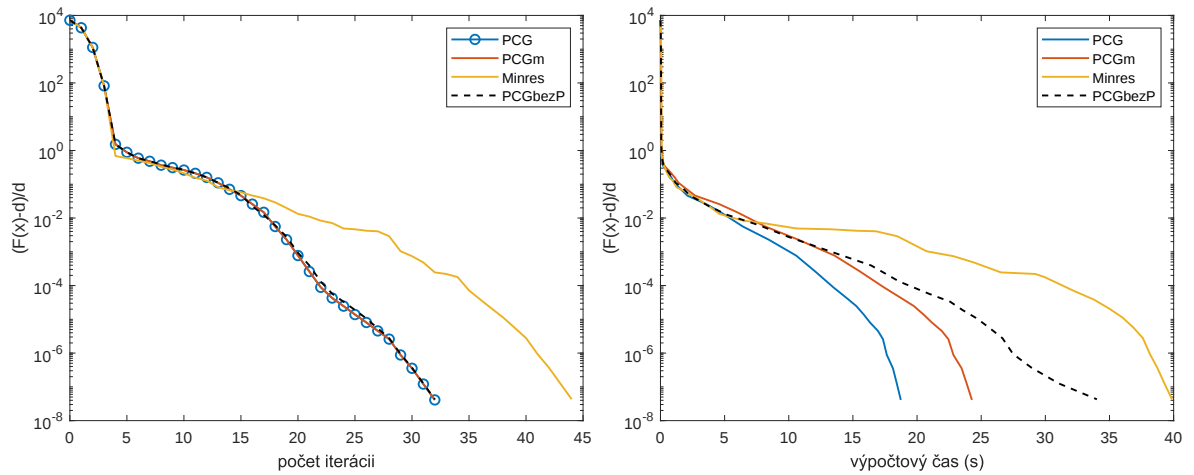


Obr. 6.6: Porovnanie výkonu metódy PD11i na náhodne generovanej úlohe (1.2) pri použití rôznych iteračných metód - naša implementácia PCG metódy, PCG metóda a Minres metóda zabudované v Matlabe, PCG metóda bez prepoďmienenia.

Tento jav je spôsobený tým, že spektrálne vlastnosti matíc Newtonových systémov sa pri zvyšovaní presnosti riešenia zhoršujú, a preto PCG bez prepoďmienenia horšie konverguje a na dosiahnutie podobne presného riešenia potrebuje stále viac iterácií.

Nová PDMVB s redukciou dimenzie

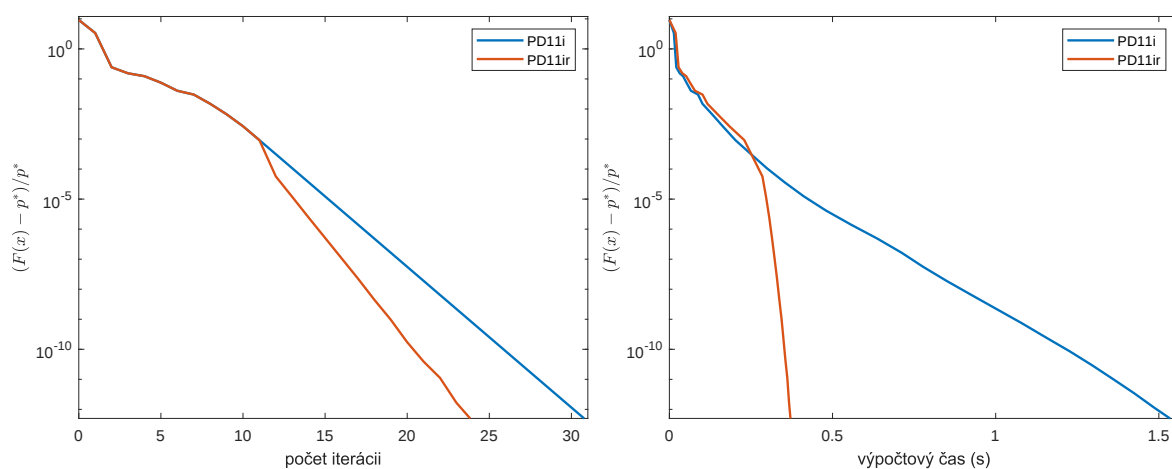
Ďalší možný spôsob ako zrýchliť výpočet primárno-duálnych smerov z Newtonových systémov sme našli vytvorením hybridnej metódy PDMVB s redukciou dimenzie. Hlavnou myšlienkou metódy je rozmuným spôsobom redukovať dimenziu dát (a tým dimenziu riešených Newtonových systémov), s ktorými metóda pracuje. V tejto časti načrt-neme fungovanie a prínosy PDMVB s redukciou dimenzie na náhodne generovanej úlohe (1.2), ktorej optimálne riešenie a hodnotu účelovej funkcie poznáme. Úloha má rozmery $m = 2400$ a $n = 12000$, optimálne riešenie obsahuje presne 1200 nenulových zložiek. Porovnanie vývoja relatívnej subtopimality medzi hybridnou metódou PD11ir s redukciou dimenzie a metódou PD11i, z ktorej PD11ir vznikla, možno vidieť na Obr. 6.8. Pri pohľade na daný priebeh je vidieť, že metódy podávajú identické riešenie až do 11-tej iterácie, kde obe dosiahli riešenie s relatívnou suboptimalitou menšou ako 10^{-3} . Metóda PD11ir potrebuje na tieto počiatočné iterácie o niečo viac výpočtového času, pretože vo svojom algoritme pomocou poznatkov o proximálnych metódach kontroluje,



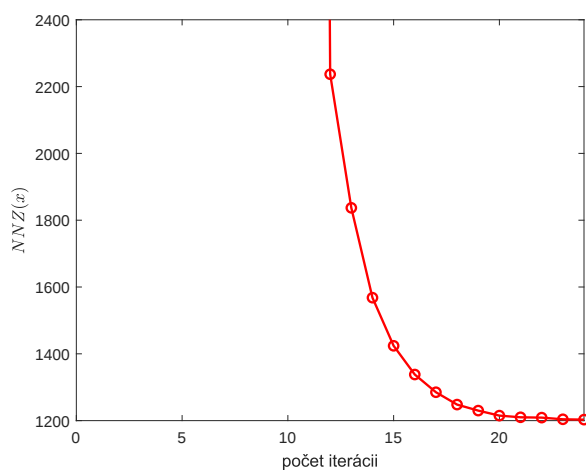
Obr. 6.7: Porovnanie výkonu metódy PD11i na TTD úlohe pri použití rôznych iteračných metód - naša implementácia PCG metódy, PCG metóda a Minres metóda zabudované v Matlabe, PCG metóda bez prepodmienenia.

či možno redukovať dimenziu dát. V nasledujúcich iteráciách sú už podmienky redukcie dimenzie (3.14), (3.15) splnené a metóda PD11ir má povolené redukovať rozmery dát, s ktorými pracuje. Týmto spôsobom metóda PD11ir sústredí optimalizáciu len na podstatné zložky riešenia, čo možno pozorovať na väčšom poklese relatívnej suboptimality a menším počtom potrebných iterácií. Taktiež sa výrazne zníži čas potrebný na výpočet primárno-duálnych smerov z Newtonových systémov, čo možno dedukovať aj z druhého grafu na Obr. 6.8. Na dosiahnutie rovnako presného riešenia s relatívnou suboptimalitou menšou ako 10^{-12} metóda PD11ir potrebuje iba 24 iterácií a $0.37s$, zatiaľ čo pôvodná metóda PD11i potrebuje 31 iterácií a $1.55s$.

Aj keď metóda PD11i dokázala nájsť veľmi presné riešenie, toto riešenie má všetky zložky nenulové. Na určenie množiny nenulových zložiek by bolo potrebné nejaké ďalšie zaokrúhľovacie pravidlo. Metóda PD11ir s redukciou dimenzie tento problém čiastočne rieši, poskytuje riedke riešenie, ktoré má 1203 nenulových zložiek. Vývoj počtu nenulových zložiek riešenia (resp. vývoj dimenzie dát) z PD11ir možno nájsť na Obr. 6.9. Metóda poskytovala do 11-tej iterácie riešenie s 12000 nenulovými zložkami, ktoré v 12-tej iterácii zredukovala na počet 2237 a v ďalších iteráciách postupne ďalej redukovala. Týmto spôsobom metóda PD11ir rieši v neskorších iteráciách algoritmu niekoľkonásobne menšie Newtonove systémy ako pôvodná metóda PD11i.



Obr. 6.8: Porovnanie vývoja relatívnej suboptimality pre metódy PD11i a PD11ir na náhodne generovanej úlohe (1.2), ktorej optimálne riešenie poznáme.



Obr. 6.9: Vývoj počtu nenulových zložiek riešenia pochádzajúceho z metódy PD11ir použitej na náhodne generovanú úlohu (1.2), ktorej optimálne riešenie má 1200 nenulových zložiek.

Základné nastavenia parametrov PDMVB

Algoritmy PDMVB majú pomerne zložitú štruktúru, obsahujú mnoho parametrov. Niektoré z nich majú na výkon týchto metód menší a iné naopak väčší vplyv. V našich experimentoch sme sa snažili udržiavať jednotlivé hodnoty parametrov PDMVB metód konštantné, ich hodnoty sme menili len sporadicky. Základné nastavenia parametrov PDMVB metód boli nasledovné:

$$\nu = 3, \quad \alpha = 0.01, \quad \beta = 0.6. \quad (6.1)$$

V prípade PDMVB, pri ktorých riešime Newtonove systémy pomocou iteračných metód, máme navyše parameter ξ z (4.46), pričom väčšina metód používa základnú hodnotu

$$\xi = 0.01. \quad (6.2)$$

A pre metódy PD12i a PD22i má tento parameter hodnotu $\xi = 0.005$.

V prípade PDMVB s redukciou dimenzie, t.j. v prípade metódy PD11ir z experimentov, potrebujeme taktiež hodnoty parametrov ϵ_{p1} a ϵ_{p2} z podmienok redukcie dimenzie (3.14), (3.15). Ich hodnoty vo väčšine experimentov volíme ako

$$\epsilon_{p1} = 0.001, \quad \epsilon_{p2} = 0.2. \quad (6.3)$$

6.2 Úloha l_1 -regulovaných najmenších štvorcov

V tejto časti vykonáme veľké porovnanie metód na umelo generovaných úlohách l_1 -regulovaných najmenších štvorcov (1.2). Budeme sledovať vplyv zmeny rôznych parametrov týchto úloh na výpočtový čas, ktorý metódy potrebujú na dosiahnutie požadovaných presností (5.3). Metódy budeme testovať na 50 náhodne generovaných úlohách každého druhu. Okrem samotného porovnania metód je našim cieľom nájsť „univerzálne najlepšie“ metódy riešenia úlohy (1.2). Z dôvodu veľkého počtu metód v tejto práci sme sa rozhodli zaviesť kvalifikáciu metód, kde metódy zredukujeme zhruba na polovicu. A takisto zavedieme do experimentov postupné vypadávanie metód. Metódy, ktorým sa pri danom type úlohy evidentne nedarí alebo ich podľa nás nemá zmysel ďalej testovať, budeme postupne vyradovať. Pre takéto metódy uvádzame v tabuľkách iba údaje z výpočtu na prvej úlohe testovacej sady.

Okrem výpočtového času metód sledujeme aj počet iterácií, ktoré metódy potrebovali na dosiahnutie riešenia s danou relatívnou supoptimalitou. Pri týchto sledovaných hodnotách v tabuľkách uvádzame minimálnu, maximálnu a priemernú nameranú hodnotu na našej testovacej vzorke 50-tich úloh.

Generovanie dát

Na generovanie umelých dát sme v práci používali niekoľko rôznych generátorov podľa potreby experimentov. Umelé dáta pre prvé experimenty (vplyv λ , m , n , $m \times n$, de) sme generovali nasledujúcim spôsobom: Najprv vygenerujeme náhodnú maticu A s rozmermi $m \times n$, s hustotou zaplnenia de a s prvkami z $[0, 1]$. Následne položíme $A = A/\sqrt{m}$ a vygenerujeme „falošné“ riešenie x_f s $0.5m$ nenulovými zložkami z intervalu $[-0.5, 0.5]$, pomocou ktorého vytvoríme vektor $b \in \mathbb{R}^m$ ako

$$b = Ax_f + e,$$

kde vektor $e \in \mathbb{R}^m$ predstavuje šum v pozorovaniach a jeho prvky sú z normálneho rozdelenia $N(0, 10^{-4})$. Základné hodnoty parametrov sú nasledovné

$$m = 2400, \quad n = 12000, \quad de = 0.05, \quad \lambda = 0.1\lambda_{max},$$

kde λ_{max} je definovaná ako (1.3). Optimálne riešenie takto generovanej úlohy nepoznáme, x_f nemá naň takmer žiadny vplyv. Pri porovnaní metód používame stopovacie kritérium (5.2). Ako štartovací bod volíme pre všetky metódy nulový vektor.

6.2.1 Kvalifikácia metód

Keďže máme naprogramovaných veľa metód a rozdiely v ich výkone sú veľké, rozhodli sme sa metódy zredukovať cca na polovicu. Metódy zredukujeme tým, že spravíme kvalifikáciu na základnej úlohe, ktorá reprezentuje „bežnú“ úlohu pri našich experimentoch. Sme si vedomí, že by sa pri niektorých menej náročných úlohách v jednotlivých experimentoch mohlo niektorej z vyradených metód dariť. No našim cieľom, okrem porovnania metód, je nájsť univerzálne najlepšie fungujúce metódy.

Kvalifikáciu metód riešenia na základnej úlohe možno nájsť v Tabuľke 6.1. Rozdiely vo výkone metód sú veľké - základné prírastkové proximal-gradientné metódy PPGd

a PPGs nedokázali dosiahnuť ani prvú požadovanú presnosť za 300s, zatiaľ čo metóda PDMVB s redukciou dimenzie PD11ir dosiahla riešenie s relatívnou suboptimalitou 10^{-9} za 0.67s. Metódy, ktoré sú označené šedou farbou, sme sa rozhodli z ďalších experimentov vyradiť. Medzi vypadnuté metódy patria aj PD11, PD12, PD21, PD22, ktoré riešia Newtonove systémy priamo. Možno tak pozorovať veľký prínos použitia iteračných metód, pretože ich verzie používajúce iteračné metódy fungovali niekoľkonásobne lepšie. Metódy PD13i a PD23i nedokázali v stanovenom čase dopočítať riešenie finálnej presnosti, pretože v neskorších fázach algoritmu pracujú s veľmi zle podmienenými maticami a iterácie ich iteračných metód prudko narastajú. Naopak, verzie týchto metód využívajúce priame metódy riešenie dopočítali do finálnej presnosti. Z Tabuľky 6.1 možno taktiež pozorovať veľký prínos modifikovaného algoritmu určovania dĺžky kroku pri metóde PGLs. Medzi vypadnuté metódy patria aj všetky prírastkové metódy MISOd, MISOs, PPGd, PPGs, PIAG, PSAG a PSAGls. Pri metódach MISOd, MISOs, PIAG, PSAG uvádzame najlepší výsledok, ktorý sme dokázali dosiahnuť voľbou dĺžky kroku tvaru $t = 2^k t_0$. Je možné, že naša implementácia prírastkových metód v Matlabe nie je optimálna aj z dôvodu indexovacieho systému prostredia Matlab. Autori týchto metód často používajú implementácie metód v jazyku C alebo metódy medzi sebou porovnávajú len cez počet iterácií, resp. počet prechodov dát. Takisto tieto metódy sú prispôbené hlavne pre prípad, keď pre rozmery úloh platí $m \gg n$, pretože ich iterácie sú nezávislé od počtu funkcií m . Z prírastkových metód najlepší výkon podala PSAG metóda.

Tabuľka 6.1: Kvalifikácia metód pre základnú úlohu (1.2) s $m = 2400$, $n = 12000$, $de = 0.05$, $\lambda = 0.1\lambda_{max}$.

Presnosť ϵ	10^{-3}		10^{-6}		10^{-9}		300s	
Metóda	čas	it.	čas	it.	čas	it.	ϵ	it.
ADMM	0,50	27	0,81	75	1,20	133		
APG	0,31	222	2,06	1475	10,64	7628		
APGls	0,58	234	4,20	1562	22,13	8272		
DPG1	0,55	222	3,57	1475	18,59	7752		
DPG1ls	0,25	92	1,22	457	4,69	1681		
DPG2	0,54	222	3,58	1475	5,51	2282		
DPG2ls	0,24	92	0,50	190	0,85	320		
DPG3	0,54	222	3,56	1475	5,89	2459		
DPG3ls	0,27	92	0,62	212	1,43	493		
MISOd	6,45	144960	21,87	482160	48,10	993360		
MISOs	4,95	108960	24,33	489600	51,57	1007520		
PD11	35,87	12	56,14	19	73,54	25		
PD11i	0,50	12	1,50	18	2,55	25		
PD11ir	0,52	12	0,58	15	0,67	20		
PD12	29,93	12	50,53	19	69,04	25		
PD12i	0,95	12	2,33	18	3,77	25		
PD13	5,87	12	9,25	19	12,16	25		
PD13i	2,94	12	94,12	19	–	–	4,2E-08	27
PD21	218,10	12	–	–	–	–	2,1E-06	17
PD21i	1,19	11	3,53	17	6,02	24		
PD22	76,82	12	119,65	18	166,62	25		
PD22i	2,03	11	5,27	18	7,94	24		
PD23	10,74	12	16,08	18	22,17	25		
PD23i	8,36	12	180,59	18	–	–	2,2E-07	20
PG	7,90	5699	35,55	25832	73,21	53199		
PGls	0,80	484	2,80	1650	5,41	3153		
PPGd	–	–	–	–	–	–	2,5E-03	11921040
PPGs	–	–	–	–	–	–	3,7E-03	11865120
PIAG	101,33	2978400	–	–	–	–	1,6E-05	8666160
PSAG	3,53	94800	15,96	422160	33,01	862560		
PSAGls	9,95	240240	46,77	1086720	99,88	2238720		

6.2.2 Globálne pozorovania z tabuliek

V ďalšej časti práce sa nachádza veľké množstvo tabuliek s výsledkami skúmaných experimentov. Keďže sme sa pri opise týchto tabuliek nechceli opakovať, rozhodli sme sa spísať pozorovania, ktoré je možné z daných tabuliek častokrát postrehnúť. Tieto pozorovania sú platné pre väčšinu tabuliek, ktoré sa v práci nachádzajú, pretože mnohokrát súvisia so stavbou algoritmov uvedených metód či s ich špecifickými vlastnosťami.

Z tabuliek možno pozorovať, že metódy DPG1ls, DPG2ls, DPG3ls s modifikovaným algoritmom určovania dĺžky kroku dosahujú výrazne lepšie výsledky ako ich varianty DPG1, DPG2, DPG3 používajúce konštantnú dĺžku kroku. Naopak metóda APGs so štandardným algoritmom určovania dĺžky kroku častokrát dosahuje redukciu v počte priemerných iterácií oproti metóde APG, avšak časová náročnosť iterácií tejto metódy je omnoho vyššia ako pri základnej APG, a preto nedosahuje redukciu vo výpočtovom čase. APGs dosahuje oproti metóde APG redukciu výpočtového času iba výnimočne, napríklad v Tabulke B.1 a B.2 z Dodatku B pre veľkú hodnotu regularizačného parametra λ .

Metódy APG, DPG1, DPG2, DPG3 (ako aj DPG1ls, DPG2ls, DPG3ls) podávajú do prvej presnosti vo väčšine tabuliek identické výsledky. Tento jav je spôsobený tým, že všetky tieto metódy sú spádovými modifikáciami APG, ktorá je do prvej presnosti spádovou na našich úlohách. Od prvého nárastu účelovej funkcie pri metóde APG začnú metódy podávať odlišné výsledky.

Takisto metóda PD11ir s redukciou dimenzie podáva do prvej presnosti podobné výsledky ako PD11i, z ktorej vznikla. V tejto fáze metóda PD11ir pre mnohé úlohy nemala ešte povolené redukovať dimenziu dát, pretože podmienky redukcie dimenzie (3.14), (3.15) neboli splnené. Pre prísnejšie presnosti býva rozdiel medzi PD11i a PD11ir veľký, možno pozorovať úsporu v počte iterácií a predovšetkým vo výpočtovom čase. Za daný jav môže práve redukcia dimenzie dát, na ktorých metóda vykonáva optimalizačný proces. Metóda PD11ir redukovaním dimenzie sústreďí optimalizáciu len na podstatné nenulové zložky riešenia.

Metódy PD11i, PD12i a PD13 pre prvú transformáciu (4.34) úloh podávajú konzistentne lepšie výsledky ako ich ekvivalenty PD21i, PD22i a PD23 pre druhú transformáciu (4.35). Tento jav je spôsobený tým, že počítane Newtonove systémy majú

pri druhej transformácii (4.35) úloh dvojnásobný rozmer ako pri prvej transformácii. Z toho dôvodu v neskorších fázach experimentov častokrát vyradíme PDMVB metódy PD21i, PD22i či PD23.

Taktiež možno pozorovať, že metódy vnútorného bodu, ktoré sú metódami druhého rádu, potrebujú na dosiahnutie presného riešenia iba desiatky iterácii bez ohľadu na úlohu. Na vypočítanie riešenia s relatívnou suboptimalitou 10^{-9} málokedy potrebovali viac ako 40 iterácii.

6.2.3 Vplyv regularizačného parametra λ

V prvom experimente skúmame vplyv zmeny hodnoty regularizačného parametra λ na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 21 hodín. Parameter λ sme volili tvaru $\gamma\lambda_{max}$, kde $\gamma \in [0.8, 0.005]$ a $\lambda_{max} = \|A^T b\|_\infty$ predstavuje hraničnú hodnotu, pri ktorej je riešením úlohy (1.2) už nulový vektor. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Hodnota regularizačného parametra λ úzko súvisí s počtom nenulových prvkov riešenia. Priemerné počty nenulových zložiek riešení 50 úloh z testovacích sád pre skúmané hodnoty regularizačného parametra λ možno nájsť v nasledujúcej tabuľke.

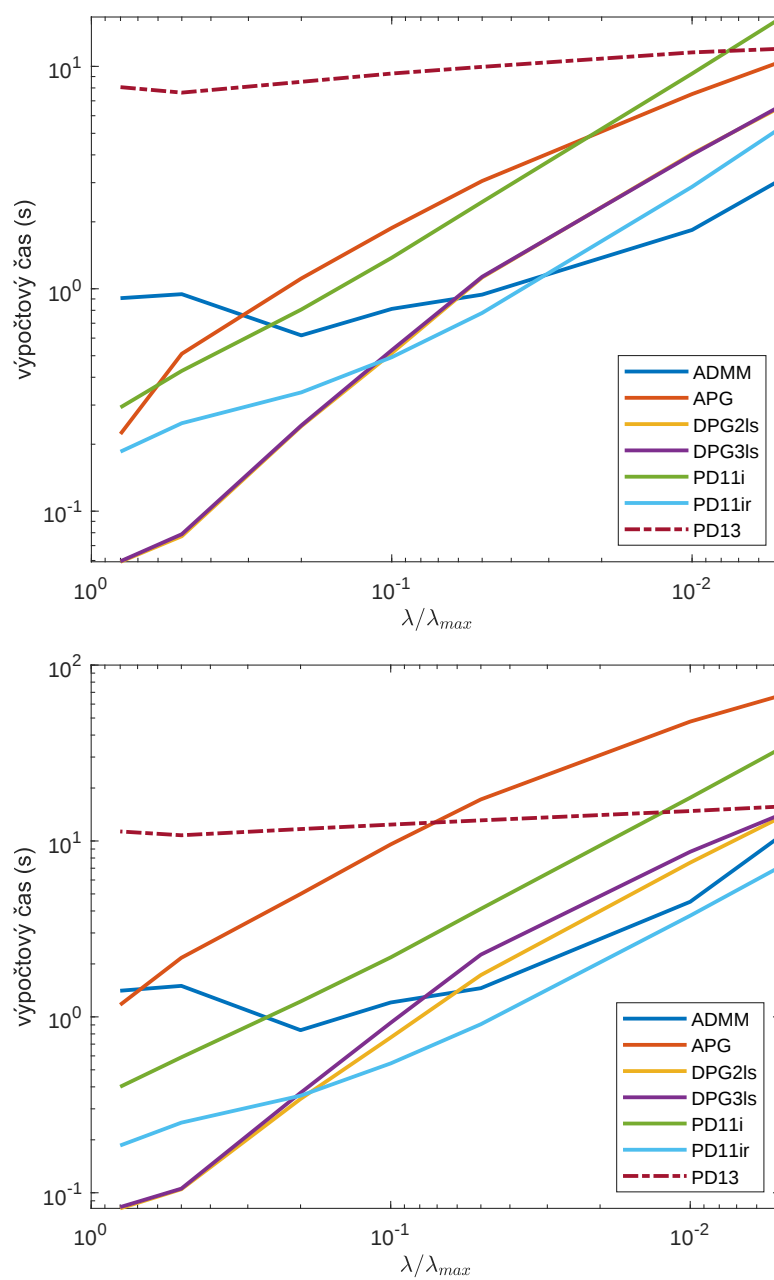
λ/λ_{max}	0.8	0.5	0.2	0.1	0.05	0.01	0.005
$NNZ(x^*)$	12	106	757	1379	1830	2278	2340

Vidíme, že pomocou zmeny hodnoty regularizačného parametra λ dokážeme riadiť počet nenulových prvkov optimálneho riešenia úloh (1.2).

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.1 až B.7 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Napríklad z Tabuliek B.1 až B.6 možno pozorovať, že výpočtový čas metódy PGls prudko narastá so zmenšovaním parametra λ , rozhodli sme sa ju ďalej netestovať. Metódy PD21i a PD22i sme sa rozhodli netestovať na posledných dvoch sádach úloh, pretože ich ekvivalenty PD11i a PD12i pre prvú transformáciu (4.34) úloh podávali konzistentne lepšie výsledky. Z Tabuliek B.1 a B.2 pre väčšie hodnoty λ možno pozorovať, že metóda APGs so štandardným algoritmom určovania dĺžky kroku podáva lepšie výsledky ako metóda APG. Tento jav je však zriedkavý z dôvodu výpočtovo náročnejších iterácii metódy APGs. Takisto možno pozorovať, že PDMVB, ktoré sú metódami druhého rádu, potrebujú na dosiahnutie presného riešenia desiatky iterácii, zatiaľ čo proximal-gradientné metódy, ktoré sú metódami prvého rádu potrebujú častokrát tisíce.

Na Obr. 6.10 možno pozorovať vplyv regularizačného parametra λ na výpočtový

čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme, že náročnosť riešených úloh prudko narastá pri znižujúcej sa hodnote parametra λ , výpočtové časy väčšiny metód prudko rastú. Zaujímavý výsledok však podáva metóda PD13, ktorá dosahuje veľmi podobné výpočtové časy pre rôzne hodnoty parametra λ . Takisto pozorujeme, že sprísnenie požadovanej presnosti má menší vplyv na výpočtový čas metódy druhého rádu PD11ir ako v prípade ostatných metód. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls a ADMM. Pre jednoduchšie úlohy s väčšou hodnotou parametra λ dosahujú najmenší výpočtový čas DPG2ls, DPG3ls a PGls. Pre náročnejšie riešiteľné úlohy s malou hodnotou λ podávajú najlepší výkon PD11ir a ADMM.



Obr. 6.10: Vplyv hodnoty regularizačného parametra λ na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2).

6.2.4 Vplyv rozmeru m

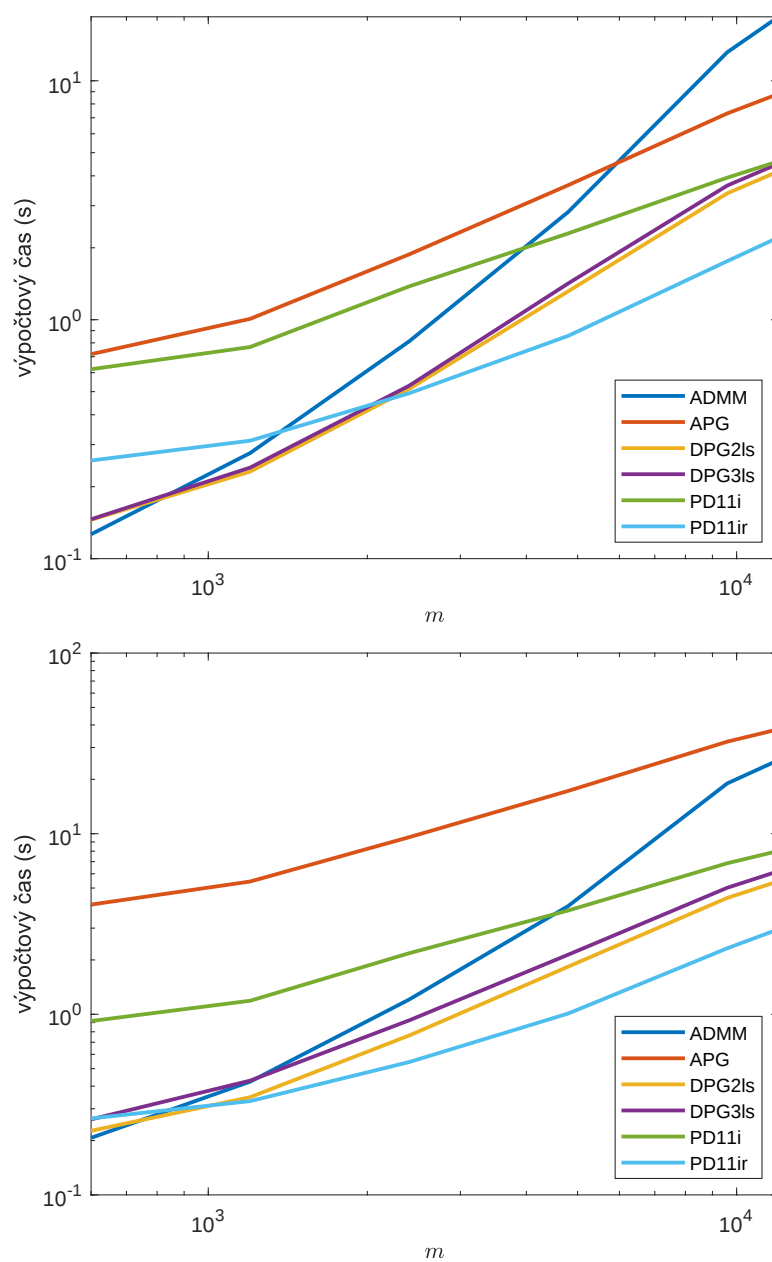
V experimente skúmame vplyv zmeny rozmeru m na výsledný výpočtový čas a počet iterácií metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 14,5h. Skúmané hodnoty rozmeru m možno nájsť v nasledujúcej tabuľke. Ostatné

m	600	1200	2400	4800	9600	12000
-----	-----	------	------	------	------	-------

hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.8 až B.13 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Všimnime si, že metódy DPG1(ls), DPG2(ls), DPG3(ls) podávajú do prvej presnosti identické výsledky, pri metódach DPG1, DPG2, DPG3 s konštantnou dĺžkou kroku je tento jav pozorovateľný aj pri relatívnej suboptimalite 10^{-6} pre väčšie hodnoty rozmeru m . Z PDMVB najlepšie obstáli PD11i a jej modifikácia PD11ir s redukciou dimenzie a z proximálnych metód najlepší výkon podali metódy DPG2ls a DPG3ls. Z tabuliek taktiež vidieť, že PD11i a PD11ir dosahujú výpočtové časy s menším rozptylom ako metódy DPG2ls a DPG3ls.

Na Obr. 6.11 možno pozorovať vplyv zmeny rozmeru m na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Prirodzene, náročnosť riešených úloh narastá so zväčšovaním rozmeru m , výpočtový čas metód rastie. Takisto pozorujeme, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas metódy APG ako v prípade PD11i a iných metód. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls. Pre jednoduchšie úlohy s menším rozmerom m dosahujú najlepší výpočtový čas ADMM, DPG2ls, DPG3ls. Pre náročnejšie riešiteľné úlohy s väčšou hodnotou m podáva jednoznačne najlepší výkon metóda PD11ir.



Obr. 6.11: Vplyv rozmeru m na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2).

6.2.5 Vplyv rozmeru n

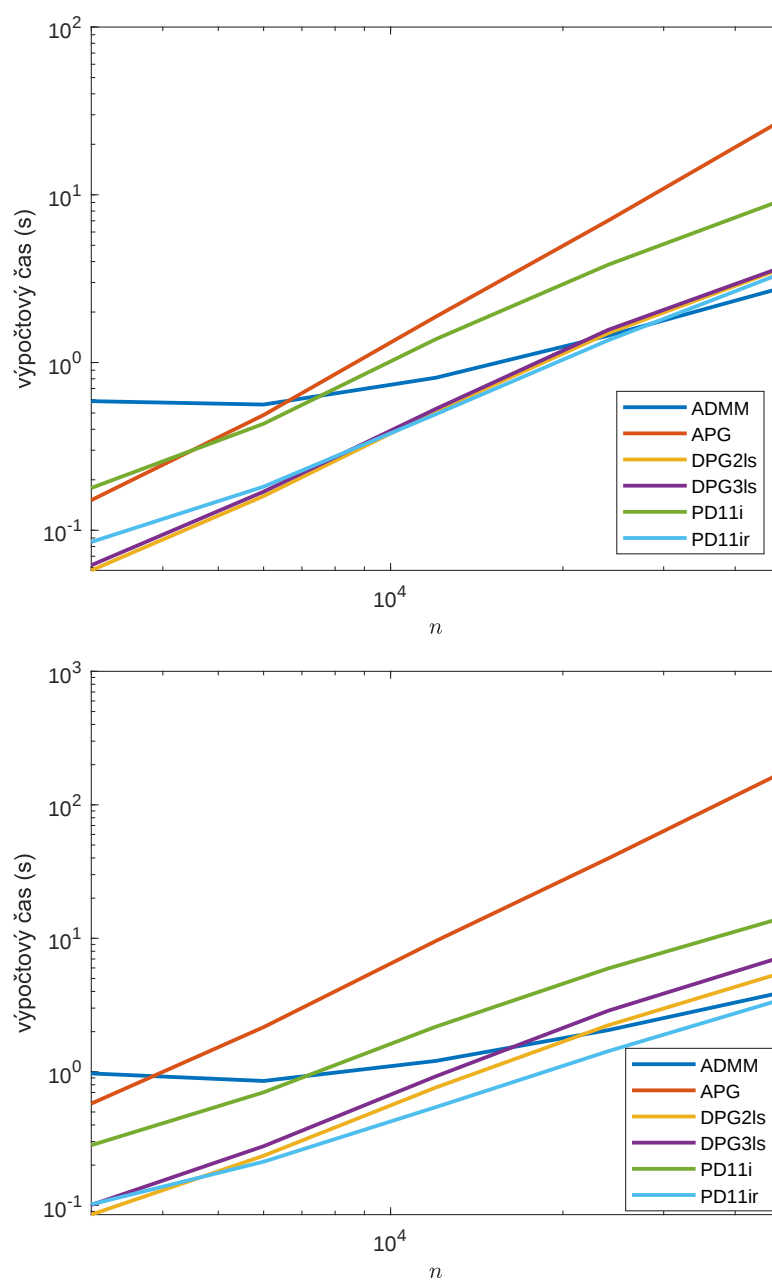
V experimente skúmame vplyv zmeny rozmeru n na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval 13 hodín. Skúmané hodnoty rozmeru n možno nájsť v nasledujúcej tabuľke. Ostatné hodnoty

n	3000	6000	12000	24000	48000
-----	------	------	-------	-------	-------

parametrov generátora úloh ostali fixné na základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.14 až B.18 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Pri relatívnej suboptimalite 10^{-6} a 10^{-9} opäť možno pozorovať veľký rozdiel medzi PD11i a PD11ir, metóda PD11ir vypočíta riešenie s relatívnou suboptilitou 10^{-9} za niekoľkonásobne rýchlejší výpočtový čas a potrebuje na to menšie množstvo iterácii. Všimnime si taktiež, že metóda PD12i podáva obstojné výsledky a je blízko metódy PD11i.

Na Obr. 6.12 možno pozorovať vplyv zmeny rozmeru n na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Prirodzene, náročnosť riešených úloh narastá so zväčšovaním rozmeru n , výpočtové časy metód rastú. Zo zobrazených metód rastie výpočtový čas metódy APG najrýchlejšie. Takisto pozorujeme, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas metódy APG prvého rádu ako pri metóde PD11i druhého rádu. Podobné pozorovanie platí pre metódy DPG2ls a DPG3ls voči metóde PDiir. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls. Pri relatívnej subotimalite 10^{-9} dosahuje najlepšie výsledky metóda PD11ir. Pre najväčiu hodnotu parametra n však veľmi dobrý výpočtový čas dosahuje metóda ADMM. Tento jav je spôsobený tým, že naša implementácia ADMM používa Choleského rozklad matice $tAA^T + I$ rozmerov $m \times m$ a výpočty čiastočne prebiehajú vektoroch rozmeru m , a preto je implementácia metódy menej závislá od rozmeru n .



Obr. 6.12: Vplyv rozmeru n na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2).

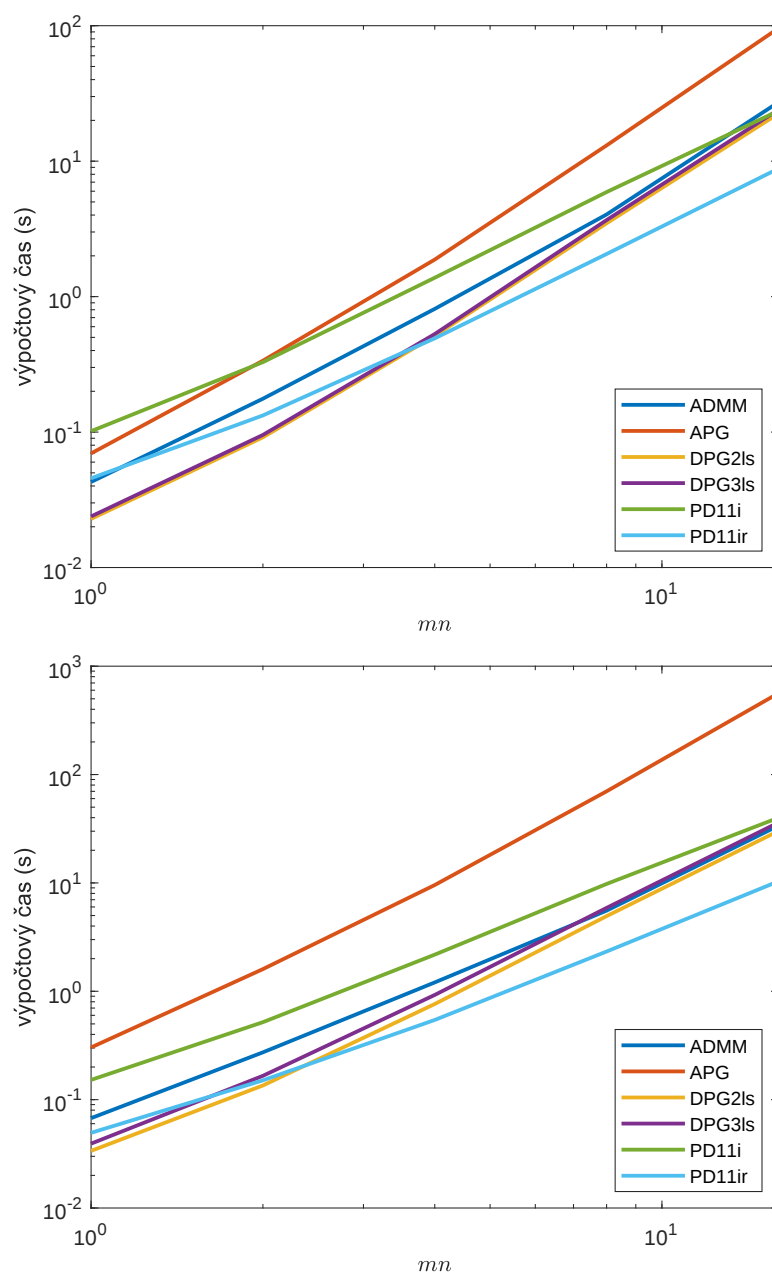
6.2.6 Vplyv rozmerov $m \times n$

V experimente skúmame vplyv zmeny oboch rozmerov m a n na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 14,5h. Rozmery $m \times n$ skúmaných úloh (1.2) sa nachádzajú v nasledujúcej tabuľke. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

m	600	1200	2400	4800	9600
n	3000	6000	12000	24000	48000

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.19 až B.23, pričom každá tabuľka prislúcha iným rozmerom skúmaných úloh a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Vo finálnej Tabuľke B.23 sme ďalej netestovali proximálne metódy APG, DPG2 a DPG3, je možné pri nich však pozorovať, že z hľadiska počtu iterácii podávali identický výkon až po presnosť 10^{-6} , pričom metóda APG má oveľa menej výpočtovo náročnú iteráciu. Pri presnosti 10^{-9} však metóda APG dosiahla už horší výpočtový čas a potrebovala oveľa viac iterácii. Z tabuliek možno vidieť, že pre relatívnu suboptimálnosť 10^{-6} a 10^{-9} je rozdiel medzi PD11i a PD11ir veľký, možno pozorovať úsporu v počte iterácii ako aj vo výpočtovom čase. V Tabuľke B.23 pre poslednú sadu úloh pozorujeme veľkú vyrovnanosť metód ADMM, DPG2ls, DPG3ls, PD11i pri presnostiach 10^{-6} a 10^{-9} .

Na Obr. 6.13 možno pozorovať vplyv zmeny oboch rozmerov m a n na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Prirodzene, náročnosť riešených úloh narastá so zväčšovaním rozmerov m a n . Výpočtové časy metód APG, DPG2ls a DPG3ls rastú rýchlejšie ako výpočtové časy metód PD11i a PD11ir. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls a ADMM. Pre jednoduchšie riešiteľné úlohy s menšími rozmermi dosahujú najlepší výpočtový čas DPG2ls, DPG3ls. Pre náročnejšie riešiteľné úlohy väčších rozmerov podáva jednoznačne najlepší výkon metóda PD11ir, pričom jej výkon sa v porovnaní s inými metódami zlepšuje pri sprísňujúcej sa presnosti. Keď metóda PD11ir dostatočne zredukuje dimenziu dát, presnejšie riešenie dopočíta veľmi rýchlo.



Obr. 6.13: Vplyv rozmerov $m \times n$ na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2).

6.2.7 Vplyv riedkosti matice dát 1

V nasledujúcom experimente skúmame vplyv riedkosti matice dát, resp. vplyv hustoty zaplnenia de matice dát, na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 34h. Skúmame hodnoty parametra de od hodnoty 0.001 až po hodnotu 1, ktorá zodpovedá plne zaplnenej matici dát. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Príprava tohto experimentu nám taktiež umožnila pozorovať, pre aké hodnoty parametra de sa oplatí pre porovnávané metódy používať matice typu sparse v prostredí Matlab. Väčšina metód má túto hraničnú hodnotu približne na úrovni $de = 0.4$. To znamená, že pre maticu zaplnenú na 40%, nezáleží, či ju pre metódy evidujeme v Matlabe ako riedku či plnú. U metód PD13 a PD23 sme pozorovali iné správanie, ich hraničná hodnota bola blízko $de = 0.05$. Pre hodnotu $de = 0.01$ bolo výhodné maticu dát evidovať ako riedku, zatiaľ čo pre hodnotu $de = 0.05$ metóda podávala lepšie výsledky pri matici, ktorú sme evidovali ako plnú. Tento zaujímavý posun oproti iným metódam je spôsobený tým, že priame PDMVB metódy PD13 a PD23 majú vo svojom algoritme násobenie matíc typu ADA^T . Pri dnešných moderných procesoroch, ktoré majú mnoho jadier je často rýchlejšie daný súčin počítať paralelizovanými algoritmami pre súčin plných matíc, t.j. „hrubou silou“, ako násobiť riedke matice. Pri násobení riedkych matíc sa veľký čas strávi hľadaním vhodných indexov nenulových prvkov, ktoré sa majú medzi sebou násobiť. Podobne Choleského rozklad predprogramovaný v Matlabe, ktorý využíva naša implementácia ADMM na rozklad matice $tAA^T + I$ na začiatku algoritmu, má hraničnú hodnotu blízko $de = 0.01$ (pri tejto hodnote bolo o niečo lepšie maticu dát evidovať ako plnú) a zvyšok metódy má hraničnú hodnotu blízko $de = 0.4$. Pre našu základnú úlohu ($de = 0.05$) je preto najvýhodnejšie použiť Choleského rozklad na plnej matici a vo zvyšku algoritmu pracovať s pôvodnou riedkou maticou. Poznamenajme, že tieto pozorovania sú do veľkej miery ovplyvnené PC zostavou, na ktorej prebiehajú výpočty. Veľký vplyv má napríklad počet jadier procesora.

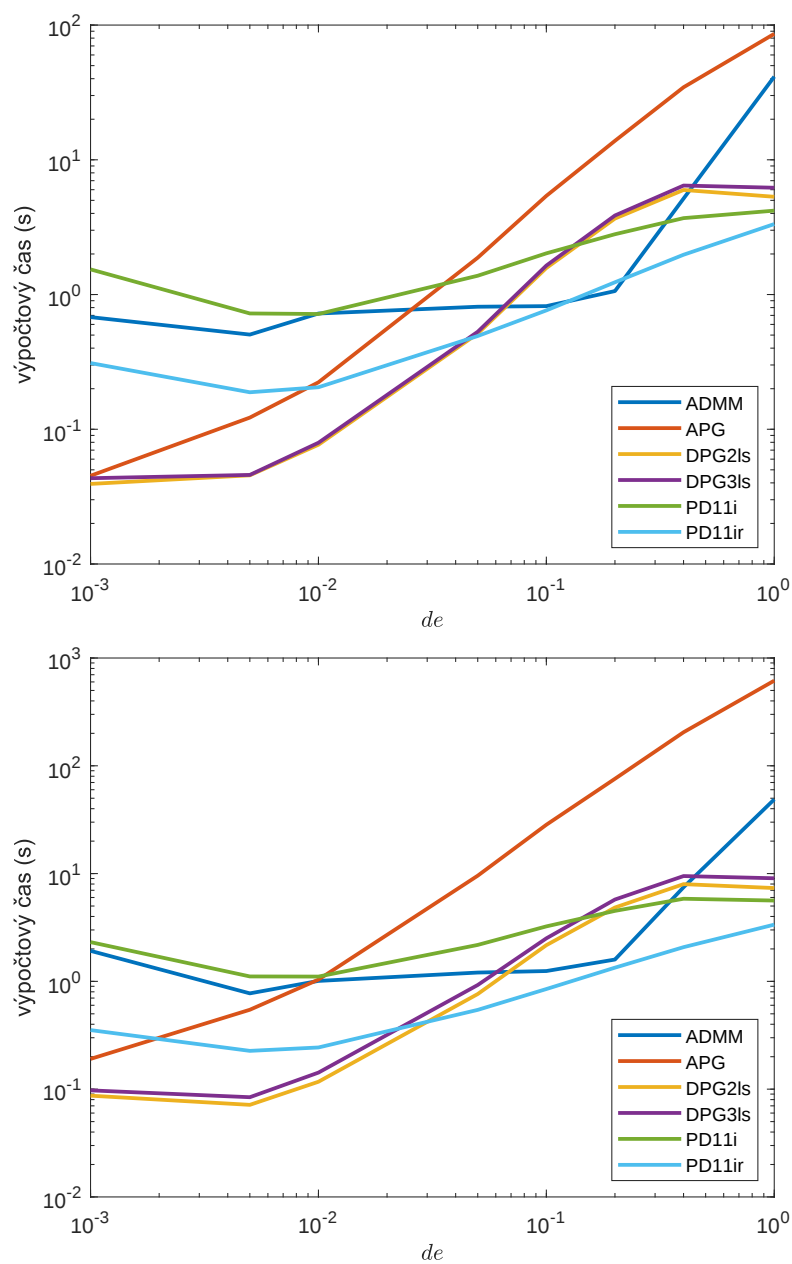
Podrobné výsledky experimentu sú uvedené v Tabuľkách B.24-B.31, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať

sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Z prvých tabuliek pozorujeme, že vysoká riedkosť dátových matíc veľmi vyhovuje proximal-gradientným metódam, predovšetkým do druhej požadovanej presnosti. Tento jav je logický, tieto metódy sú prvého rádu a majú značne jednoduchšiu štruktúru iterácii ako PDMVB. Výpočtový čas metódy APG však prudko narastá so zväčšujúcou hustotou zaplnenia matice dát. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach. Avšak niektoré výsledky z tabuliek sú mátuce. Napríklad, pri porovnaní výsledkov metód medzi prvými Tabuľkami B.24 a B.25 pozorujeme zlepšenie výsledkov napr. pre metódy ADMM, DPG2ls, DPG3ls, PD11i, PD11ir a zhoršenie výsledkov pre APG, DPG2, DPG3, PD13. Podobne pri porovnaní výsledkov metód medzi poslednými Tabuľkami B.30 a B.31 pozorujeme zlepšenie výsledkov u DPG2ls, DPG3ls, PD11i a zhoršenie výsledkov pre metódy ADMM, APG, DPG2, DPG3, PD11ir, PD13. Tieto pozorovania sú prítomné aj na Obr. 6.14.

Na Obr. 6.14 možno pozorovať vplyv zmeny hustoty zaplnenia de matice dát na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme rozdielne chovanie metód so zväčšovaním parametra de . APG je jediná metóda na Obr. 6.14, ktorej výpočtový čas so zväčšovaním parametra de neustále rastie. PDMVB metódy PD11i a PD11ir sú zmenou parametra de ovplyvnené menej ako iné metódy, čo sa dá povedať aj o metóde ADMM až po poslednú sledovanú hodnotu parametra de . Takisto pozorujeme, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas metódy APG ako v prípade PD11i, PD11ir a iných metód. Pre úlohy s menšou hodnotou parametra de dosahujú najlepší výpočtový čas metódy DPG2ls, DPG3ls. Pre úlohy s väčšou hodnotou parametra de podáva najlepší výkon metóda PD11ir.

Z výsledkov experimentu sme však nadobudli pocit, že sa niečo deje s dátami, o čom zatiaľ netušíme. Pri následnom skúmaní vlastností dát sme zistili, že aj keď držíme parametre generátora úloh na fixných hodnotách, tak sa podstatne mení počet nenulových zložiek riešení daných úloh, čo možno pozorovať v nasledujúcej tabuľke.

de	0.001	0.005	0.01	0.05	0.1	0.2	0.4	1
$NNZ(x^*)$	1507	1600	1553	1379	1219	994	491	181



Obr. 6.14: Vplyv hustoty zaplnenia matice dát de na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2).

6.2.8 Vplyv riedkosti matice dát 2

V predošlom experimente sme zistili, že pri zvyšovaní počtu nenulových prvkov matice A (resp. hustoty zaplnenia de) sa menil počet nenulových prvkov optimálneho riešenia, čo nám menilo náročnosť riešených úloh.

V prípade úlohy l_1 -regulovaných najmenších štvorcov (1.2) je však možné vytvoriť generátor dát, pri ktorom optimálne riešenie úlohy poznáme. Existuje viacero spôsobov, ako sa dajú takéto úlohy generovať, všetky sú založené na prispôbení dát A , b , x^* tak, aby boli splnené podmienky optimality úlohy (1.2), t. j.

$$0 \in A^T(Ax^* - b) + \lambda \partial \|x^*\|_1. \quad (6.4)$$

Takéto generátory možno nájsť napríklad v publikáciách [78], [36], [71].

Pretože by nám generátor tohto typu umožnil vykonať ďalšie zaujímavé experimenty, rozhodli sme sa taký generátor naprogramovať a použiť.

Generovanie dát

Umelé dáta generujeme príbuzným spôsobom ako v [78], ale tento postup sme upravili pre potreby našich experimentov. Ako prvé zvolíme počet nenulových zložiek $p_n \leq m$ optimálneho riešenia x^* a vygenerujeme náhodnú maticu B s rozmermi $m \times n$, s hustotou zaplnenia de a s prvkami z $[0, 1]$. Následne vytvoríme vektor $v \in \mathbb{R}^m$, ktorého zložky sú z rovnomerného rozdelenia na intervale $[0, 1]$. Vektor v znormujeme na jednotkovú dĺžku a v ďalšom kroku zoradíme zložky vektora $B^T v$ zostupne podľa ich absolútnych hodnôt. Pre zjednodušenie značenia predpokladajme, že to bolo pôvodné usporiadanie. V ďalšom kroku položíme $A = BD$, kde D je diagonálna škálovacia matica, pričom jej diagonálne zložky d_{ii} sú vytvorené ako

$$d_{ii} = \begin{cases} \frac{\lambda}{|b_i^T v|} & \text{pre } i = 1, \dots, p_n \\ 1 & \text{ak } i > p_n \text{ a } |b_i^T v| \leq 0.1\lambda, \\ \frac{\zeta_i \lambda}{|b_i^T v|} & \text{inak} \end{cases}$$

kde ζ_i sú z rovnomerného rozdelenia na intervale $[0, 1]$. Optimálne riešenie úlohy generujeme spôsobom

$$x_i^* = \begin{cases} (dr + (1 - dr)\zeta_i) \operatorname{sgn}(a_i^T v) & \text{pre } i = 1, \dots, p_n, \\ 0 & \text{inak} \end{cases},$$

pričom ζ_i sú z rovnomerného rozdelenia na intervale $[0, 1]$ a premenná $dr \leq 1$ riadi tzv. dynamický rozsah riešenia. Následne optimálne riešenie upravíme na požadovanú dĺžku ρ (resp. vzdialenosť od počiatku)

$$x^* = \rho \frac{x^*}{\|x^*\|_2}.$$

V poslednom kroku generátora dopočítame vektor $b \in \mathbb{R}^m$ pomocou vzťahu

$$b = v + Ax^*.$$

Všimnime si, že v danom generátore je hodnota regularizačného parametra λ potrebná predtým, ako poznáme hraničnú hodnotu $\lambda_{max} = \|A^T b\|_\infty$. Základné hodnoty parametrov sme sa snažili nastaviť tak, aby boli v súlade s ich hodnotami pri úlohách pochádzajúcich z predošlého generátora. Základné hodnoty parametrov sú nasledovné

$$m = 2400, \quad n = 12000, \quad de = 0.05, \quad \lambda = 0.05, \quad p_n = 1200, \quad \rho = 10, \quad dr = 0.001.$$

Ako štartovací bod volíme pre všetky metódy nulový vektor. Keďže optimálne riešenie takto generovanej úlohy poznáme, pri porovnaní metód používame stopovacie kritérium (5.1).

Nový generátor nám taktiež umožní vykonať ďalšie zaujímavé experimenty ako skúmanie vplyvu počiatocnej vzdialenosti od optimálneho riešenia, vplyvu počtu nenulových prvkov optimálneho riešenia a vplyvu dynamického rozsahu optimálneho riešenia.

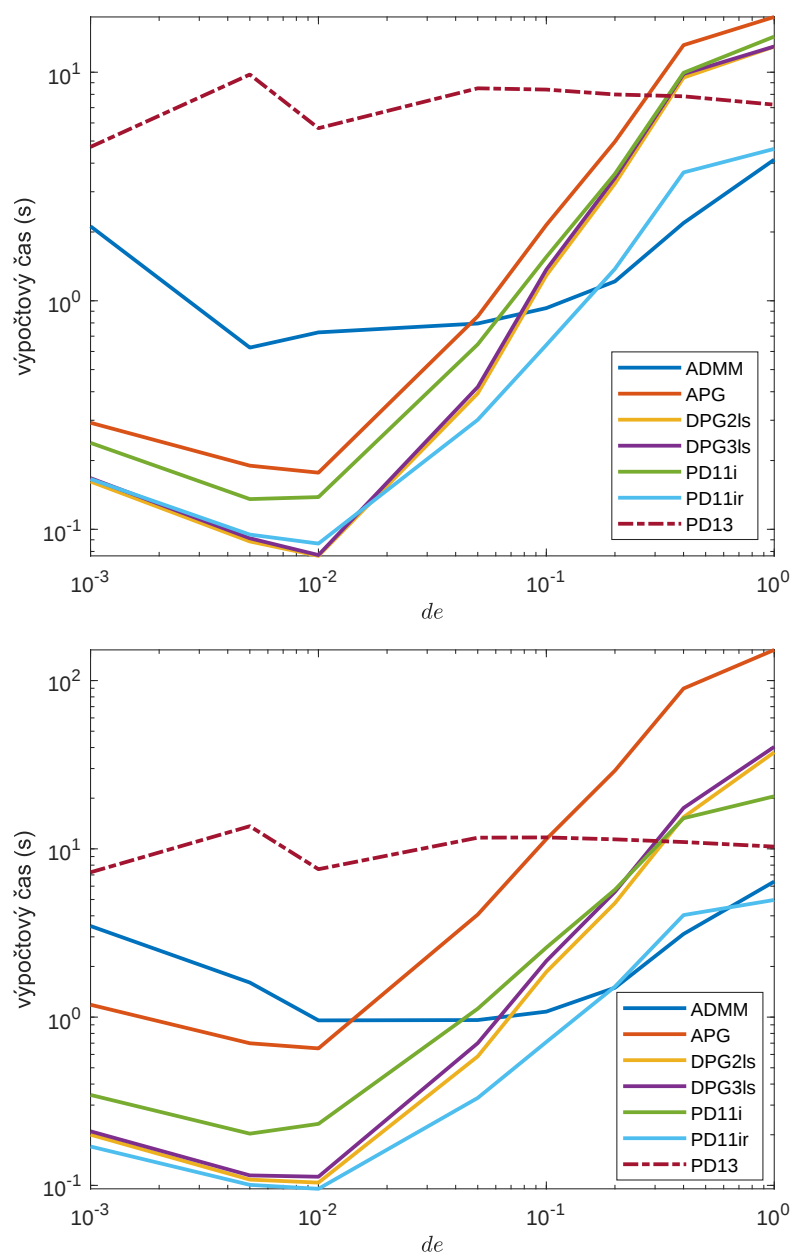
Vplyv riedkosti matice dát

V nasledujúcom experimente skúmame vplyv riedkosti matice dát, resp. vplyv hustoty zaplnenia de matice dát, na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 29h. Skúmame hodnoty parametra de od hodnoty 0.001 až po hodnotu 1, ktorá zodpovedá plne zaplnenej

matici dát. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.32 až B.39 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v kapitole o globálnych pozorovaniach. Z prvých 3 tabuliek však možno pozorovať neočakávaný výkon metód, pre väčšinu metód sa výpočtový čas so zväčšovaním parametra de znižuje. Po bližšom skúmaní tohto javu sme zistili, že prvé dve sady úloh majú maticu dát s výrazne odlišnými spektrálnymi vlastnosťami ako úlohy prislúchajúce hodnotám hustoty zaplnenia $de \geq 0.01$. Tento jav teda súvisí so spôsobom generovania matic dát A a je pozorovateľný aj na Obr. 15. V Tabuľkách B.34 až B.39 sa metódy správajú v súlade s očakávaním.

Na Obr. 6.15 možno pozorovať vplyv zmeny hustoty zaplnenia de matice dát na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme, že náročnosť riešených úloh narastá so zväčšovaním parametra de pre hodnoty $de \geq 0.01$. Správanie metód pre nižšie hodnoty parametra de bolo do značnej miery ovplyvnené výrazne inými vlastnosťami generovaných matic. Metódy PD13 a PD23 sú zmenou parametra de ovplyvnené menej ako iné metódy. Tento jav súvisí s tým, že maticu dát pre tieto metódy evidujeme ako plnú pre hodnoty parametra $de \geq 0.05$. Metóda ADMM podáva veľmi dobré výsledky pre hustotu zaplnenia $de \geq 0.1$. V experimente celkovo najlepší výkon dosiahla PDMVB metóda PD11ir, metóda podala najkonzistentnejšie výsledky. Pri všetkých nastaveniach parametra de a presnostiach 10^{-6} a 10^{-9} má kompetitívny výpočtový čas.



Obr. 6.15: Vplyv hustoty zaplnenia matice dát de na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2), ktorých riešenie poznáme.

6.2.9 Vplyv vzdialenosti od optimálneho riešenia

V experimente skúmame vplyv počiatkovej vzdialenosti metód od optimálneho riešenia na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Ako štartovací bod volíme pre všetky metódy nulový vektor. Výpočet experimentu trval viac ako 10.5 hodín. Skúmané hodnoty parametra $\rho = \|x^* - x_0\|_2$, možno nájsť v nasledujúcej tabuľke. Ostatné hodnoty parametrov generátora úloh ostali fixné na

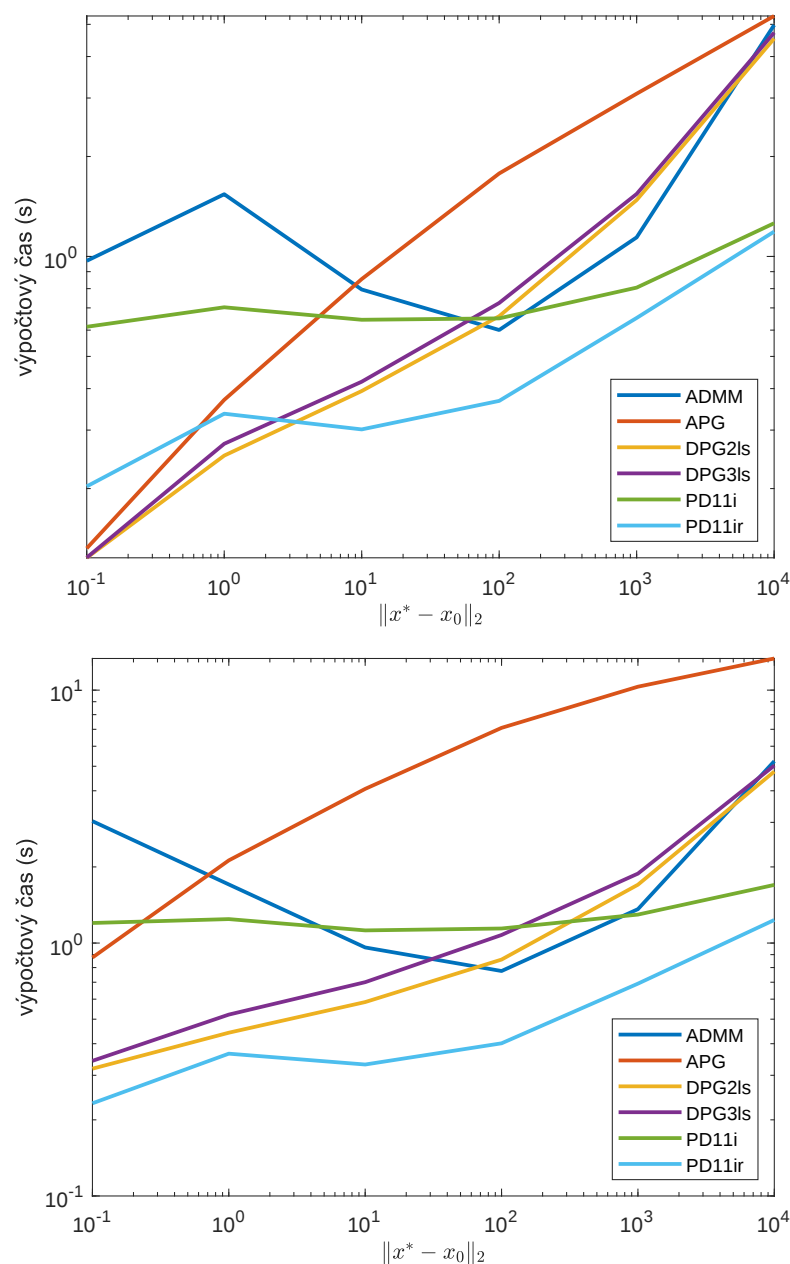
ρ	0.1	1	10	100	1000	10000
--------	-----	---	----	-----	------	-------

základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.40 až B.45 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Z tabuliek pozorujeme, že výpočtový čas metódy PGls s rastúcou vzdialenosťou prudko narastá. Podobné pozorovanie platí aj pre metódu APG, tento jav však nie je prekvapením, pretože hodnota ρ sa nachádza priamo v konvergenčnom výsledku metódy (Veta 2.2.2.). V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek. V Tabuľke B.45 pozorujeme pomerne vyrovnané výsledky metód PD11i a PD11ir až po druhú pozorovanú presnosť 10^{-6} , z toho usudzujeme, že na daných úlohách má metóda PD11ir povolené redukovat' dimenziu pomerne neskoro alebo dimenziu dát redukuje len v malej miere. Pre relatívnu suboptimalitu 10^{-9} je už možné pozorovať prínosy metódy PD11ir v podobe menšieho počtu potrebných iterácii ako aj lepšieho výpočtového času.

Na Obr. 6.16 možno pozorovať vplyv zmeny počiatkovej vzdialenosti metód od optimálneho riešenia na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme, že potrebný výpočtový čas proximal-gradientných metód APG, DPG2 a DPG3 narastá so zväčšovaním počiatkovej vzdialenosti ρ . Naopak, PDMVB metóda PD11i na zmenu počiatkovej vzdialenosti takmer nereaguje. Jej modifikácia PD11ir s redukciou dimenzie na zmenu tohto parametra reaguje viac, avšak napriek tomu stále prináša oproti pôvodnej metóde PD11i

zrýchlenie výpočtového času. Takisto pozorujeme, že sprísnenie požadovanej presnosti má najväčší vplyv na výpočtový čas APG oproti ostatným metódam na Obr. 6.16. Naopak, pre metódu PD11ir má sprísnenie požadovanej presnosti najmenší vplyv. V experimente celkovo najlepšie obstála metóda PD11ir. Pre relatívnu suboptimalitu riešení 10^{-9} metóda PD11ir nemá konkurenciu a pri relatívnej suboptimalite 10^{-6} podávajú lepší výsledok len proximal-gradientné metódy avšak iba 2 najmenšie počiatočné vzdialenosti.



Obr. 6.16: Vplyv počiatocnej vzdialenosti od optimálneho riešenia $\rho = \|x^* - x_0\|_2$ na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2), ktorých riešenie poznáme.

6.2.10 Vplyv počtu nenulových prvkov optimálneho riešenia

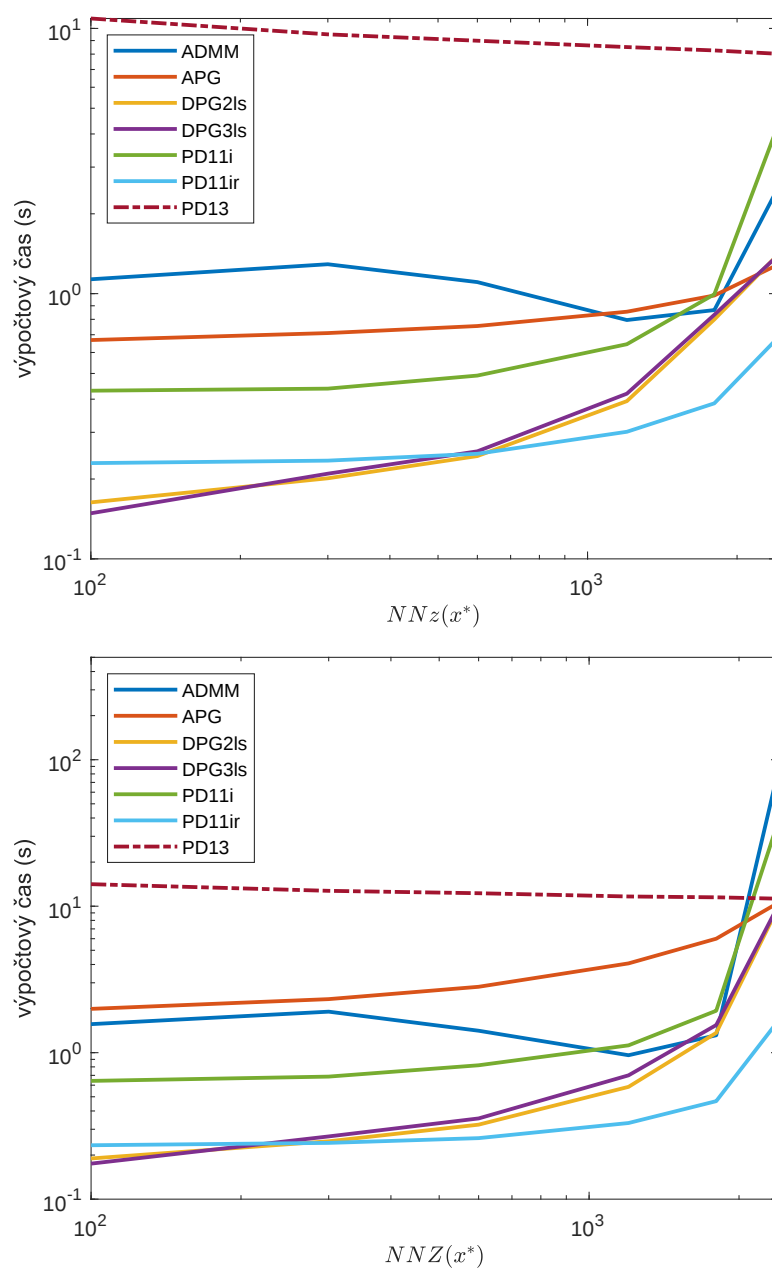
V prvom experimente práce sme testovali vplyv zmeny regularizačného parametra λ na výkon metód, pričom sme poukázali na fakt, že parameter λ riadi počet nenulových zložiek optimálneho riešenia úloh (1.2). Počet nenulových prvkov optimálneho riešenia sme však vedeli ovládať len čiastočne, medzi generovanými úlohami z jednotlivých sád sa táto hodnota vedela líšiť. V nasledujúcom experimente skúmame vplyv počtu nenulových prvkov optimálneho riešenia na výsledný výpočtový čas a počet iterácií metód pre 3 požadované presnosti (5.3). Regularizačný parameter λ ako aj iné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách. Výpočet experimentu trval viac ako 11.5 hodín. Skúmané počty nenulových prvkov optimálneho riešenia možno nájsť v nasledujúcej tabuľke.

$NNZ(x^*)$	100	300	600	100	1200	1800	2400
------------	-----	-----	-----	-----	------	------	------

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.46 až B.51 z Dodatku B, pričom každá tabuľka prislúcha inému počtu nenulových prvkov optimálneho riešenia a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek. Takisto si možno všimnúť, že metódy dosahujú veľmi konzistentné výsledky, hodnoty priemerného, maximálneho a minimálneho výpočtového času sú pri jednotlivých metódach pomerne blízko seba. Výnimkou je posledná Tabuľka B.51 pre najväčší počet nenulových zložiek optimálneho riešenia, úlohy z tejto sady boli výrazne ťažšie riešiteľné. Pri mnohých metódach vidíme výrazný nárast výpočtového času. V tejto tabuľke pri relatívnej suboptimalite 10^{-6} a 10^{-9} pozorujeme veľmi vyrovnaný výkon metód APG, DPG1ls, DPG2ls a DPG3ls. Riešenie s relatívnou suboptimalitou 10^{-9} metóda PD11ir dosiahla za viac ako 20-krát rýchlejší výpočtový čas ako pôvodná metóda PD11i.

Na Obr. 6.17 možno pozorovať vplyv počtu nenulových prvkov optimálneho riešenia na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme, že výpočtový čas väčšiny metód narastá

pri zväčšujúcom sa počte nenulových prvkov optimálneho riešenia úloh a taktiež, že najväčší nárast výpočtového času sa dosahuje pri maximálnom počte nenulových prvkov optimálneho riešenia. Zaujímavý výsledok však podávajú PDMVB metódy PD13 a PD23, ktoré dosahujú veľmi podobné výpočtové časy pre rôzne hodnoty počtu nenulových prvkov optimálneho riešenia - veľmi podobné pozorovanie sme mali aj pri prvom experimente, kde sme pozorovali vplyv zmeny regularizačného parametra λ na výkon metód. V experimente celkovo najlepšie výpočtové časy dosahovali metódy PD11ir, DPG2ls a DPG3ls. Pre jednoduchšie riešiteľné úlohy s malým počtom nenulových prvkov riešenia dosiahli najnižší výpočtový čas metódy DPG2ls a DPG3ls. Ťažšie riešiteľné úlohy s väčším počtom nenulových prvkov optimálneho riešenia vyriešila najrýchlejšie metóda PD11ir.



Obr. 6.17: Vplyv počtu nenulových prvkov optimálneho riešenia na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2), ktorých riešenie poznáme.

6.2.11 Vplyv dynamického rozsahu optimálneho riešenia

V článku [71] autori ukázali, že pri ich spôsobe generovania dát má veľký vplyv na výkon metód tzv. dynamický rozsah optimálneho riešenia. Prirodzene nás teda zaujímalo, aký vplyv má tento parameter na výkon nami skúmaných metód pri našom spôsobe generovania úloh (1.2). Dynamický rozsah riešenia je definovaný [71] ako

$$\Theta(x^*) = \frac{\max\{|x^*| : x^* \neq 0\}}{\min\{|x^*| : x^* \neq 0\}}. \quad (6.5)$$

V našom generátore úloh túto veličinu riadi parameter dr , pričom platí $\Theta(x^*) \approx 1/dr$.

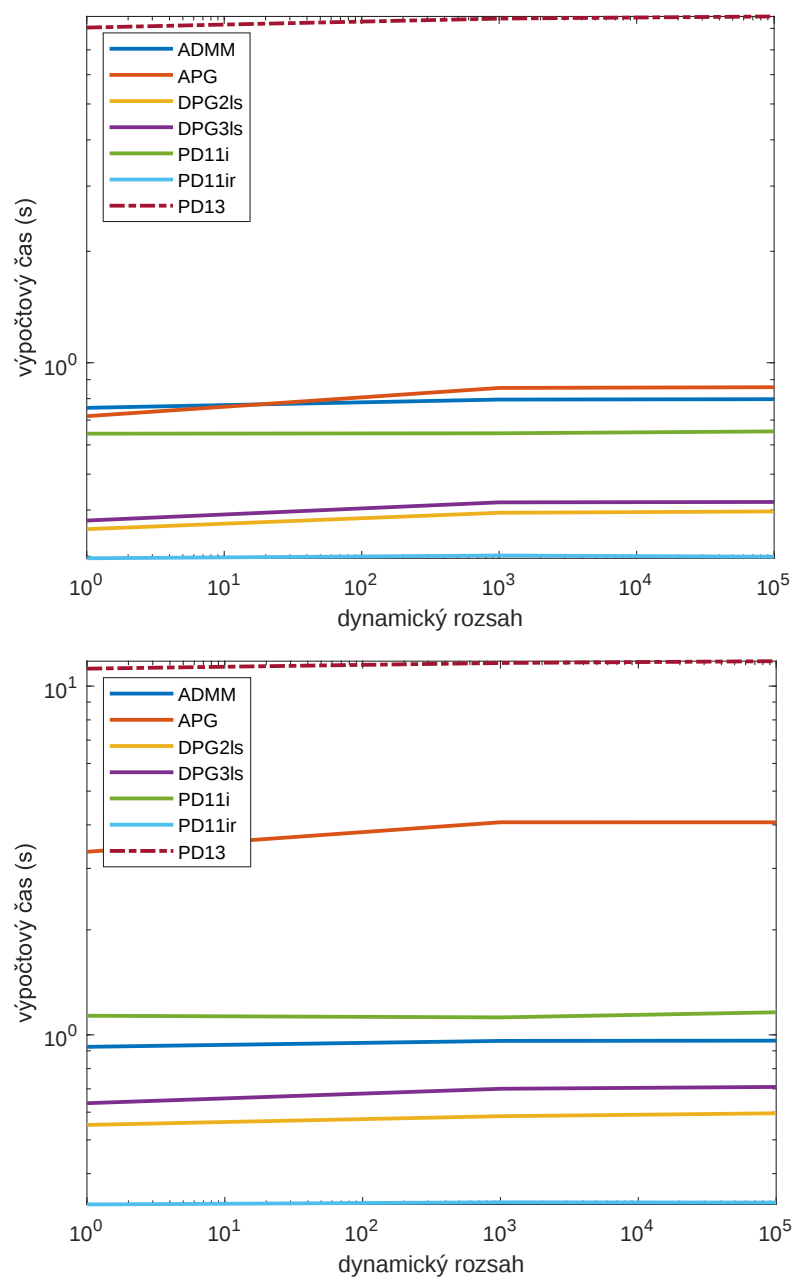
V experimente skúmame vplyv parametra dr na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 3.5 hodiny. Skúmané hodnoty parametra dr , možno nájsť v nasledujúcej tabuľke. Ostatné

dr	1	10^{-3}	10^{-5}
------	---	-----------	-----------

hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.52 až B.54 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek. Pri porovnaní výsledkov jednotlivých metód medzi tabuľkami sa len ťažko hľadajú rozdiely. Výpočtový čas väčšiny metód jemne rastie. Najväčší rozdiel vo výpočtových časoch je medzi prvými dvoma tabuľkami pri metódach APG, APGls, DPG1, DPG1ls a PGLs.

Na Obr. 6.17 možno pozorovať vplyv zmeny dynamického rozsahu optimálneho riešenia na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme veľmi malý nárast výpočtového času metód pri zväčšujúcom sa dynamickom rozsahu optimálneho riešenia. Pre naše metódy a náš spôsob generovania dát má dynamický rozsah optimálneho riešenia minimálny vplyv na výpočtový čas metód. Opäť možno pozorovať, že sprísnenie požadovanej presnosti spôsobí výraznejšie zhoršenie výpočtového času u metódy APG ako u iných metód. Táto zmena naopak najviac vyhovuje metóde PD11ir. Riešenie s relatívnou suboptimálnosťou 10^{-6} a 10^{-9} za najlepší výpočtový čas vypočíta metóda PD11ir. Dobrý výkon dosiahli aj metódy DPG2ls a DPG3ls.



Obr. 6.18: Vplyv dynamického rozsahu optimálneho riešenia na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2), ktorých riešenie poznáme.

6.2.12 Škálovateľnosť

Na skúmanie vplyvu veľkosti úloh na výpočtový čas metód bol v [56] vykonaný test škálovateľnosti, preto na našich skúmaných metódach vykonáme podobný test.

Generovanie dát

Podobne ako v [56], sme umelé dáta pre tento experiment generovali spôsobom: Najprv vygenerujeme náhodnú maticu A s rozmermi $m \times n$, pričom rozmer $m = 0.1n$ a matica má okolo $30m$ nenulových prvkov (hustota zaplnenia $de = 30/n$), ktoré sú nezávislé identicky rozdelené zo štandardného normálneho rozdelenia $N(0, 1)$. Následne vygenerujeme „falošné“ riešenie x_f s $0.25n$ nenulovými zložkami z intervalu $[-0.5, 0.5]$, pomocou ktorého vytvoríme vektor $b \in \mathbb{R}^m$ ako

$$b = Ax_f + e,$$

kde vektor $e \in \mathbb{R}^m$ predstavuje šum v pozorovaniach a jeho prvky sú z normálneho rozdelenia $N(0, 10^{-4})$. Optimálne riešenie takto generovanej úlohy nepoznáme. Pri porovnaní metód používame stopovacie kritérium (5.2). Ako štartovací bod volíme pre všetky metódy nulový vektor. Narozdiel od [56] budeme metódy testovať na 50-tich náhodne generovaných úlohách každého typu.

Kvalifikácia metód

Keďže sa výraznejšie zmenil spôsob generovania dát, vykonali sme taktiež novú kvalifikáciu metód na úlohe z prostredného experimentu s rozmermi $m = 5000$, $n = 50000$ a regularizačným parametrom $\lambda = 0.1\lambda_{max}$.

Pri primárno-duálnych metódach vnútorného bodu sme použili základné nastavenia (6.1), (6.2) a (6.3), pričom bola pre všetky metódy použitá hodnota $\xi = 0.002$.

V Tabuľke 6.2 sa nachádza kvalifikácia metód riešenia na základnej úlohe. Rozdiely vo výkone metód sú veľké. Metódy PD11, PD21, PPGd, PPGs nedokázali za poskytnutý čas 300s nájsť ani riešenie s relatívnou suboptimalitou 10^{-3} . Metóda PD21, ktorá rieši Newtonove systémy priamo, nedokázala vypočítať ani jednu iteráciu z dôvodu zaplnenia operačnej pamäte počítača. Metóda PD11, ktorá vo svojom algoritme rieši Newtonov systém menších rozmerov, vypočítala za čas 300s iba 7 iterácii. Opäť možno

teda pozorovať veľký prínos tzv. nepresných PDMVB, metódy PD11i a PD21i úlohu bez problémov vyriešili. Metódy, ktoré sú označené šedou farbou, sme sa rozhodli z ďalších experimentov vyradiť. Keďže sa rozmery riešených úloh budú ďalej aj zväčšovať, nemá zmysel testovať presné PDMVB, t.j. PD11, PD12, PD13, PD21, PD22 a PD23. Medzi vypadnuté metódy patria aj všetky prírastkové metódy MISOd, MISOs, PPGd, PPGs, PIAG, PSAG a PSAGls, ktoré na danej úlohe nepodávali dobrý výkon. Pri metódach MISOd, MISOs, PIAG, PSAG uvádzame najlepší výsledok, ktorý sme dokázali dosiahnuť voľbou dĺžky kroku tvaru $t = 2^k t_0$.

Pri porovnaní výsledkov z predošlými experimentami môžeme pozorovať, že daný spôsob generovania dát výrazne napomáha proximal-gradientným metódam, všetky z nich sa výrazne zlepšili. Metóda DPG2ls poskytla riešenie s relatívnou suboptimálnosťou 10^{-9} len za 0.18s.

Tabuľka 6.2: Kvalifikácia metód pre základnú úlohu (1.2) s $m = 5000$, $n = 50000$, $de = 6E - 04$, $\lambda = 0.1\lambda_{max}$.

Presnosť ϵ	10^{-3}		10^{-6}		10^{-9}		$300s$	
Metóda	čas	it.	čas	it.	čas	it.	ϵ	it.
ADMM	0,88	23	1,98	77	3,56	155		
APG	0,02	31	0,10	152	0,40	605		
APGls	0,04	30	0,19	147	0,74	584		
DPG1	0,04	31	0,18	152	0,71	605		
DPG1ls	0,03	19	0,12	85	0,38	278		
DPG2	0,04	31	0,18	152	0,33	273		
DPG2ls	0,03	19	0,10	74	0,18	131		
DPG3	0,04	31	0,18	152	0,34	290		
DPG3ls	0,03	19	0,11	78	0,22	153		
MISOd	46,51	434160	–	–	–	–	3,4E-06	2827200
MISOs	74,17	690720	–	–	–	–	2,2E-05	2802960
PD11	–	–	–	–	–	–	1,6E-01	7
PD11i	0,21	14	0,42	20	0,77	27		
PD11ir	0,24	14	0,28	16	0,29	20		
PD12	29,48	14	49,28	20	69,78	26		
PD12i	0,45	14	0,94	20	1,76	27		
PD13	25,28	14	36,05	20	55,19	26		
PD13i	1,30	18	44,13	25	–	–	3,4E-08	44
PD21	–	–	–	–	–	–	–	–
PD21i	0,53	13	1,06	19	1,88	26		
PD22	60,38	12	116,80	19	168,76	25		
PD22i	0,74	13	1,93	20	3,73	27		
PD23	21,70	12	34,34	19	53,52	25		
PD23i	2,02	18	75,83	25	–	–	6,6E-08	34
PG	0,09	138	0,96	1477	2,16	3356		
PGls	0,04	37	0,23	272	0,48	591		
PPGd	–	–	–	–	–	–	6,4E-03	5786400
PPGs	–	–	–	–	–	–	1,2E-02	5724720
PIAG	16,05	261600	172,00	2791200	–	–	1,8E-07	4860480
PSAG	16,69	271200	169,86	2788560	–	–	1,7E-07	4907040
PSAGls	52,61	771840	–	–	–	–	5,6E-06	4404720

Vplyv rozmernosti úloh

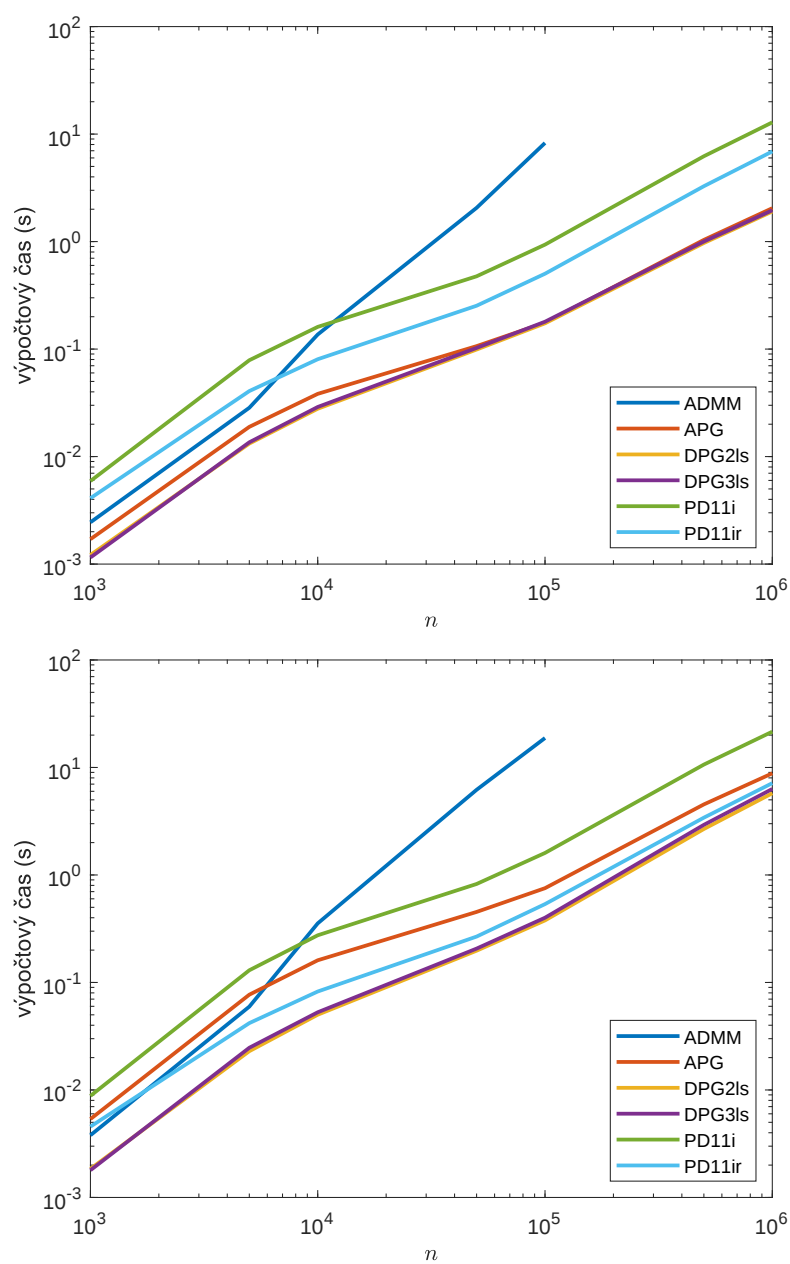
V experimente skúmame vplyv rozmernosti úloh na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Ako štartovací bod volíme pre všetky metódy nulový vektor. Výpočet experimentu trval viac ako 6.5 hodiny. Skúmame vplyv rozmernosti úloh riadenej parametrom n pre hodnoty z nasledujúcej tabuľky. Ostatné

n	1000	5000	10000	50000	100000	500000	1000000
-----	------	------	-------	-------	--------	--------	---------

hodnoty parametrov generátora úloh sú odvodené z hodnoty rozmeru n .

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.55 až B.61 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra n a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Z tabuliek pozorujeme, že výpočtový čas metódy ADMM s rastúcou rozmernosťou úloh prudko narastá a pri matici s rozmermi 50000×500000 metóda zlyhala z dôvodu zaplnenia operačnej pamäte počítača pri počítaní Choleského rozkladu matice $tAA^T + I$. Z tabuliek vidíme, že dané dáta vyhovujú najmä proximal-gradientným metódam, pričom najlepší výkon podávajú metódy DPG2ls, DPG3ls, DPG1ls a APG. Z primárno-duálnych metód vnútorného bodu majú najlepšie výpočtové časy metódy PD11i a PD11ir.

Na Obr. 6.19 možno pozorovať vplyv rozmernosti úloh riadenej parametrom n na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Náročnosť riešených úloh narastá so zväčšovaním rozmernosti úloh. Pozorujeme, že výpočtové časy ADMM rastú rýchlejšie ako pri iných metódach. V experimente celkovo najlepšie obstáli metódy DPG2ls, DPG3ls, APG, DPG1ls a PD11ir. Takisto vidíme, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas metód prvého rádu APG, DPG2ls, DPG3ls ako v prípade metódy druhého rádu PD11ir. Metóda PD11ir pri výpočte riešení s relatívnou suboptimalitou 10^{-9} dosahuje kompetitívne výpočtové časy najmä pre najrozmernejšie úlohy.



Obr. 6.19: Vplyv rozmernosti úlohy na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.2).

6.3 Úloha l_1 -regulovanej logistickej regresie

V tejto časti vykonáme veľké porovnanie metód na umelo generovaných úlohách l_1 -regulovanej logistickej regresie (1.7). Budeme sledovať vplyv zmeny rôznych parametrov týchto úloh na výpočtový čas a počet iterácií, ktorý metódy potrebujú na dosiahnutie presností (5.3). Metódy budeme testovať na 50 náhodne generovaných úlohách každého druhu. Okrem samotného porovnania metód je našim cieľom nájsť „univerzálne najlepšie“ metódy riešenia úlohy (1.7). Metodika testovania bude rovnaká ako pri predošlej úlohe (1.2). Najprv použijeme kvalifikáciu metód na zredukovanie množstva porovnávaných metód a neskôr aplikujeme taktiež vypadávanie metód. Pre vyradené metódy uvádzame v tabuľkách iba údaje z výpočtu na prvej úlohe testovacej vzorky.

Generovanie dát

Keďže chceme v tejto práci poskytnúť aj určité porovnanie náročnosti riešiteľnosti úloh (1.2) a (1.7), rozhodli sme sa použiť analogické nastavenia hlavných parametrov úloh a generovať maticu dát A podobným spôsobom. Umelé dáta pre väčšinu experimentov v tejto časti sme generovali nasledujúcim spôsobom: Najprv vygenerujeme náhodnú maticu A s rozmermi $m \times n$ s hustotou zaplnenia de a s prvkami z intervalu $[0, 1]$. Následne vygenerujeme „falošné“ riešenie x_f s $0.5m$ nenulovými zložkami z intervalu $[-0.5, 0.5]$ a $y_f \in [-0.5, 0.5]$, pomocou ktorých vytvoríme vektor skupín $b \in \mathbb{R}^m$ ako

$$b = \text{sgn}(Ax_f + y_f \mathbf{1} + e),$$

kde vektor $e \in \mathbb{R}^m$ predstavuje šum a jeho prvky sú z normálneho rozdelenia $N(0, 10^{-2})$. Predošlé generovanie falošného riešenia (x_f, y_f) a vektora skupín b vykonávame, až pokiaľ je počet pozorovaní s $b_i = 1$ z intervalu $[0.975m/2, 1.025m/2]$, aby sme mali početnosť skupín s pozitívnym a negatívnym výsledkom približne rovnakú. Predchádzajúci spôsob generovania vektora skupín b bol inšpirovaný lineárnym klasifikátorom (1.6). Základné hodnoty parametrov sú nasledovné

$$m = 2400, \quad n = 12000, \quad de = 0.05, \quad \lambda = 0.1\lambda_{max}.$$

Optimálne riešenie takto generovanej úlohy nepoznáme, a preto pri porovnaní metód používame stopovacie kritérium (5.2). Ako štartovací bod volíme $x_0 = 0_n$, $y_0 = 0$.

6.3.1 Kvalifikácia metód

Podobne ako v predošlej časti, zredukujeme počet porovnávaných metód na polovicu pomocou kvalifikácie metód. Metódy zredukujeme tým, že spravíme kvalifikáciu na základnej úlohe, ktorá reprezentuje „bežnú“ úlohu pri našich experimentoch.

Výsledky kvalifikácie sa nachádzajú v Tabulke 6.3. V tabulke možno pozorovať veľké rozdiely vo výkone metód. Prírastkové proximal-gradientné metódy PPGd, PPGs a PDMVB metóda PD21 nedokázali dosiahnuť ani prvú požadovanú presnosť za 300s, zatiaľ čo PDMVB metóda s redukciou dimenzie PD11ir vypočítala riešenie s relatívnou suboptimalitou 10^{-9} za 0.9s. Metódy, ktoré sú označené šedou farbou, sme sa rozhodli z ďalších experimentov vyradiť. Medzi vypadnuté metódy patria aj PD11, PD12, PD21, PD22, ktoré riešia Newtonove systémy priamo. Verzie týchto metód, ktoré naopak riešia Newtonove systémy pomocou iteračných metód (tzv. nepresné PDMVB), fungovali omnoho lepšie. Napríklad, metóda PD11i úlohu vyriešila za 4.19s a metóda PD11 za 177s. Metódy PD13i a PD23i nedokázali v stanovenom čase dopočítať riešenie s relatívnou suboptimalitou 10^{-9} pretože v neskorších fázach algoritmu pracujú s veľmi zle podmienenými maticami a iterácie ich iteračných metód tak prudko narastajú. Naopak, verzie týchto metód, počítajúce Newtonove systémy priamo, riešenie dopočítali do presnosti 10^{-9} . Z Tabuľky 6.3 možno taktiež pozorovať veľký prínos modifikovaného algoritmu určovania dĺžky kroku pri metódach PGLs, DPG1ls, DPG2ls a DPG3ls. Medzi vypadnuté metódy patria aj všetky prírastkové metódy MISOd, MISOs, PPGd, PPGs, PIAG, PSAG a PSAGls. Pri metódach MISOd, MISOs, PIAG, PSAG uvádzame najlepší výsledok, ktorý sme dokázali dosiahnuť voľbou dĺžky kroku tvaru $t = 2^k t_0$. Pri metóde MISOs a predovšetkým PSAG môžeme pozorovať mimoriadne veľký výkonnostný posun pri použití náhodnosti. Tieto metódy sme však napriek obstojnému výkonu vyradili, pretože daný spôsob určovania dĺžky kroku nevyhovuje ďalším experimentom. Navyše, uvedeným spôsobom určovania dĺžky kroku síce nájdeme hodnotu, pri ktorej metódy dosahujú najnižší výpočtový čas, avšak často sa nám podarilo nájsť takú dĺžku kroku, ktorá fungovala len pri „dobrej náhodnosti“. Pri „zlej náhodnosti“ metódy MISOs a PSAG divergovali. Ak by sme pri metóde PSAG použili overenú dĺžku kroku (4.32) či jej 16-násobok, ako odporúčajú autori SAG metódy, tak v tom prípade už metóda nedosahuje dobré výsledky - presnosť 10^{-9} dosiahne až za 99.5s a viac.

Tabuľka 6.3: Kvalifikácia metód pre základnú úlohu (1.7) s $m = 2400$, $n = 12000$, $de = 0.05$, $\lambda = 0.1\lambda_{max}$.

Presnosť ϵ	10^{-3}		10^{-6}		10^{-9}		300s	
Metóda	čas	it.	čas	it.	čas	it.	ϵ	it.
APG	0,42	294	2,98	2120	17,27	12427		
APGls	0,71	251	4,85	1753	26,86	9757		
DPG1	0,72	294	5,22	2120	30,22	12337		
DPG1ls	0,29	94	1,56	530	6,43	2186		
DPG2	0,73	294	5,22	2120	8,46	3422		
DPG2ls	0,29	94	1,11	370	1,49	497		
DPG3	0,73	294	5,21	2120	8,89	3611		
DPG3ls	0,29	94	1,17	392	1,84	620		
MISOd	19,56	431760	35,01	773520	60,60	1340160		
MISOs	3,42	73920	11,74	254640	23,92	521280		
PD11	94,81	14	135,62	20	177,22	26		
PD11i	0,60	14	2,60	20	4,19	27		
PD11ir	0,52	14	0,80	18	0,90	22		
PD12	22,13	14	37,43	20	53,26	26		
PD12i	1,53	14	4,52	20	6,55	27		
PD13	8,56	14	12,04	20	15,77	26		
PD13i	11,40	13	–	–	–	–	4,1E-06	20
PD21	–	–	–	–	–	–	9,0E-02	8
PD21i	1,65	14	6,58	20	10,09	27		
PD22	44,70	13	81,82	20	114,72	26		
PD22i	4,51	14	11,18	20	15,14	26		
PD23	13,97	13	21,33	20	27,61	26		
PD23i	26,23	13	–	–	–	–	3,6E-05	17
PG	14,79	10734	73,41	53360	152,90	111198		
PGls	1,19	646	4,49	2457	8,70	4768		
PPGd	–	–	–	–	–	–	2,5E-03	11156772
PPGs	–	–	–	–	–	–	3,7E-03	11002246
PIAG	19,37	593280	95,67	2941440	199,63	6144720		
PSAG	1,88	55680	5,86	176640	12,00	360240		
PSAGls	9,68	250800	46,76	1228800	97,10	2559120		

6.3.2 Vplyv regularizačného parametra λ

V prvom experimente skúmame vplyv zmeny hodnoty regularizačného parametra λ na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 16.5 hodín. Parameter λ sme volili tvaru $\gamma\lambda_{max}$, kde $\gamma \in [0.8, 0.01]$ a λ_{max} je daná vzťahom (1.8), pričom predstavuje hraničnú hodnotu, pri ktorej je riešením úlohy (1.7) už nulový vektor váh $x^* = 0$. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Hodnota regularizačného parametra λ úzko súvisí s počtom nenulových prvkov riešenia. Priemerné počty nenulových zložiek riešení 50 úloh z testovacích sád pre skúmané hodnoty regularizačného parametra λ možno nájsť v nasledujúcej tabuľke.

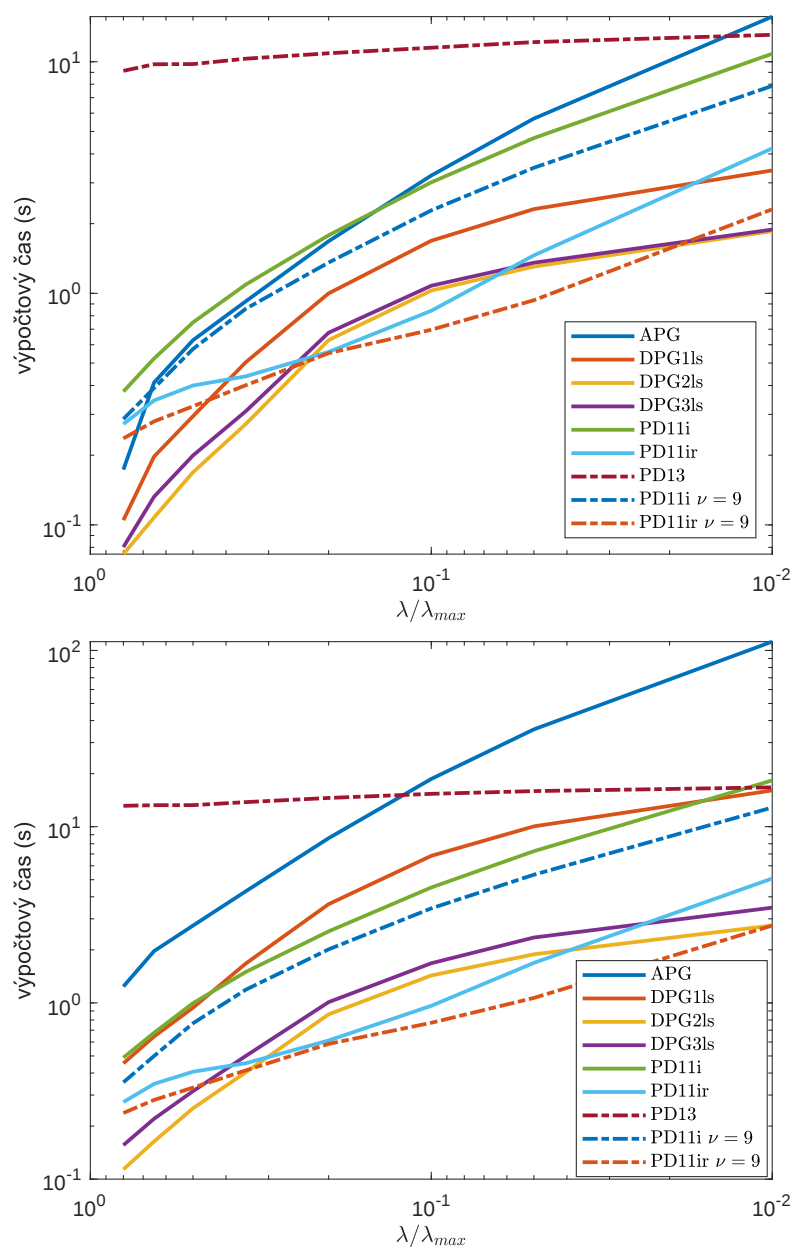
λ/λ_{max}	0.8	0.65	0.5	0.35	0.2	0.1	0.05	0.01
$NNZ(x^*)$	13	59	203	512	986	1333	1508	1693

Vidíme, že pomocou zmeny hodnoty regularizačného parametra λ dokážeme riadiť počet nenulových prvkov optimálneho riešenia úloh (1.7). Preto môžeme tento experiment vnímať aj ako test vplyvu počtu nenulových zložiek optimálneho riešenia na výkon metód. Poznamenajme, že pre hodnoty regularizačného parametra $\lambda \leq 0.01\lambda_{max}$ už pri našom spôsobe generovania dát dostatočne nepribúdali nenulové zložky optimálneho riešenia, a preto sme sa rozhodli na týchto úlohách metódy ďalej netestovať.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.62 až B.69 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra λ a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Napríklad z Tabuliek B.62 až B.69 možno pozorovať, že výpočtový čas metód APGLs a DPG1 narastá so znižovaním parametra λ viac ako u iných metód. V Tabuľke B.67 tieto metódy potrebujú na výpočet riešenia s relatívnou sub-optimalitou 10^{-9} oveľa viac výpočtového času ako iné metódy, a preto sme sa rozhodli tieto metódy ďalej netestovať. Metódy PD21i a PD22i sme sa rozhodli netestovať na poslednej sade úloh, pretože ich ekvivalenty PD11i a PD12i pre prvú transformáciu (4.34) úloh podávali konzistentne lepšie výsledky. Z Tabuliek B.62 až B.65 pre väčšie hodnoty λ možno pozorovať, že metóda APGLs so štandardným algoritmom určovania

dĺžky kroku podáva lepšie výsledky ako metóda APG. Tento jav je však zriedkavý z dôvodu výpočtovo náročnejších iterácií metódy APGs. Takisto možno pozorovať, že PDMVB, ktoré sú metódami druhého rádu, potrebujú na dosiahnutie presného riešenia desiatky iterácií, zatiaľ čo proximal-gradientné metódy, ktoré sú metódami prvého rádu potrebujú častokrát tisíce iterácií. Pri metódach DPG1ls, DPG2ls a DPG3ls pozorujeme veľký prínos modifikovaného algoritmu určovania dĺžky kroku, oproti metódam s konštantnou dĺžkou kroku DPG1, DPG2, DPG3 dosahujú niekoľkonásobne rýchlejší výpočtový čas. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek pri úlohe (1.2), pretože mnohé súvisia so stavbou či fungovaním algoritmov metód.

Na Obr. 6.20 možno pozorovať vplyv regularizačného parametra λ na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Okrem metód, ktorých výsledky uvádzame v Tabuľkách B.62 až B.69, sa tu nachádzajú aj PDMVB metódy PD11i a PD11ir pre iné nastavenie parametra ν . Pri danom nastavení obe metódy dosahujú lepšie výsledky ako so základným nastavením (6.1) pre úlohy v tomto experimente. Pozorujeme, že náročnosť riešených úloh narastá pri znižujúcej sa hodnote parametra λ (a pri zväčšujúcom sa počte nenulových prvkov optimálneho riešenia). Zaujímavý výsledok však podáva metóda PD13 (a takisto PD23), ktorej výpočtové časy rastú len pomalým tempom pre rôzne hodnoty parametra λ . Identické pozorovanie sme uskutočnili pri skúmaní vplyvu hodnoty regularizačného parametra λ pre úlohu (1.2). Takisto pozorujeme, že sprísnenie požadovanej presnosti má menší vplyv na výpočtový čas metód druhého rádu PD11i, PD11ir, PD13 ako v prípade metód APG, DPG1ls, ktoré sú metódami prvého rádu. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls a DPG3ls. Pre jednoduchšie úlohy s väčšou hodnotou parametra λ dosahujú najmenší výpočtový čas DPG2ls a DPG3ls. Pre náročnejšie riešiteľné úlohy s malou hodnotou λ podávajú najlepší výkon PD11ir, DPG2ls a DPG3ls.



Obr. 6.20: Vplyv hodnoty regularizačného parametra λ na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.7).

6.3.3 Vplyv rozmeru m

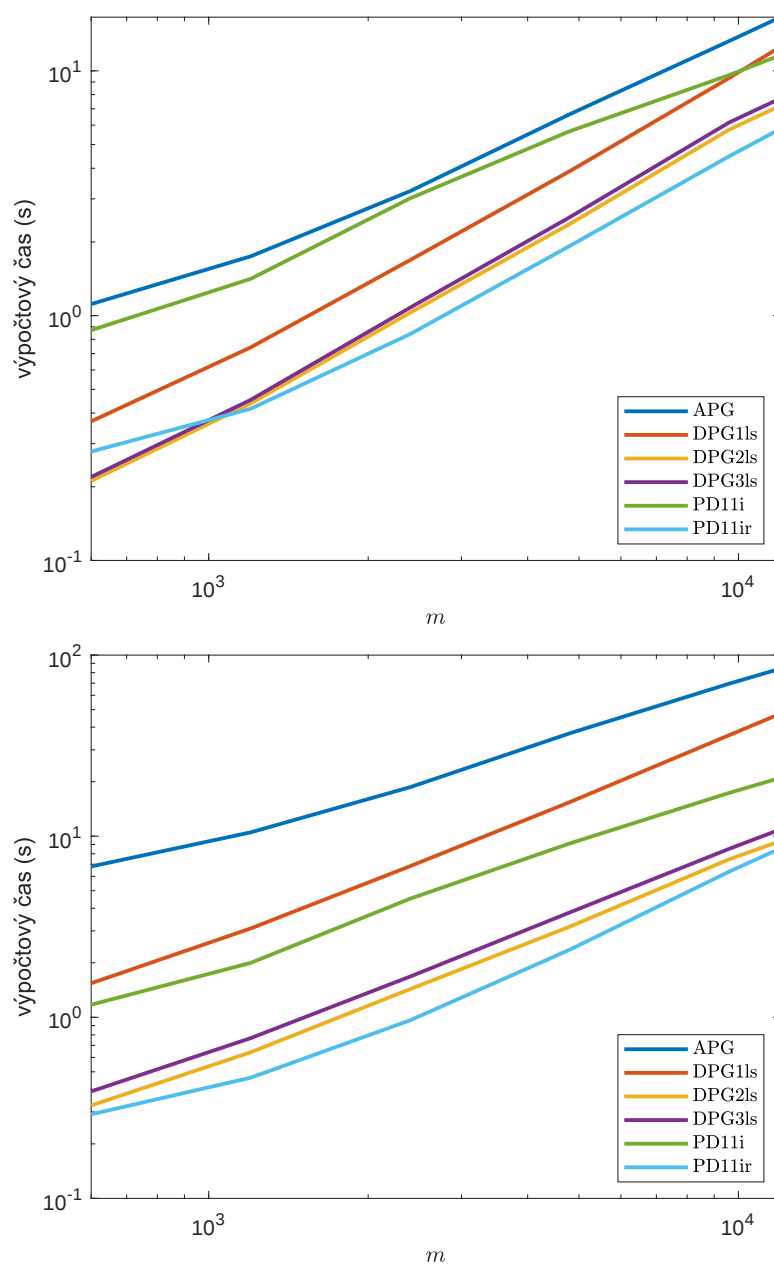
V experimente skúmame vplyv zmeny rozmeru m na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval 26 hodín. Skúmané hodnoty rozmeru m možno nájsť v nasledujúcej tabuľke. Ostatné hodnoty

m	600	1200	2400	4800	9600	12000
-----	-----	------	------	------	------	-------

parametrov generátora úloh ostali fixné na základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.70 až B.75 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Všimnime si, že metódy DPG2(ls), DPG3(ls) podávajú do prvej presnosti identické výsledky. Pri metódach APG, DPG2, DPG3 s konštantnou dĺžkou kroku je tento jav pozorovateľný aj pri presnosti 10^{-6} pre najväčšiu hodnotu rozmeru m . Tento jav je spôsobený tým, že metóda APG je do tejto presnosti na úlohách spádovou. Môžeme vďaka tomu pozorovať, o koľko spomalilo metódy DPG2 a DPG3 počítanie a porovnávanie funkčných hodnôt. Pri relatívnej suboptimalite 10^{-9} však už tieto spádové modifikácie dosahujú lepšie výpočtové časy ako APG. Z PDMVB najlepšie obstála PD11ir s redukciou dimenzie a z proximal-gradientných metód najlepší výkon podali metódy DPG2ls a DPG3ls. Z tabuliek taktiež vidieť, že PD11ir dosahuje výpočtové časy s menším rozptylom ako metódy DPG2ls a DPG3ls.

Na Obr. 6.21 možno pozorovať vplyv zmeny rozmeru m na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Prirodzene, náročnosť riešených úloh narastá so zväčšovaním rozmeru m , výpočtové časy metód rastú. Takisto pozorujeme, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas metód APG, DPG1ls ako v prípade PD11i. Pri porovnaní metód PD11i a PD11ir pozorujeme veľkú úsporu vo výpočtovom čase, ktorú sme dosiahli vďaka redukovaniu dimenzie. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls, pričom PD11ir má o niečo lepšie výpočtové časy vo všetkých sadách úloh pri presnostiach 10^{-6} a 10^{-9} okrem prvej sady úloh pri presnosti 10^{-6} .



Obr. 6.21: Vplyv rozmeru m na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.7).

6.3.4 Vplyv rozmeru n

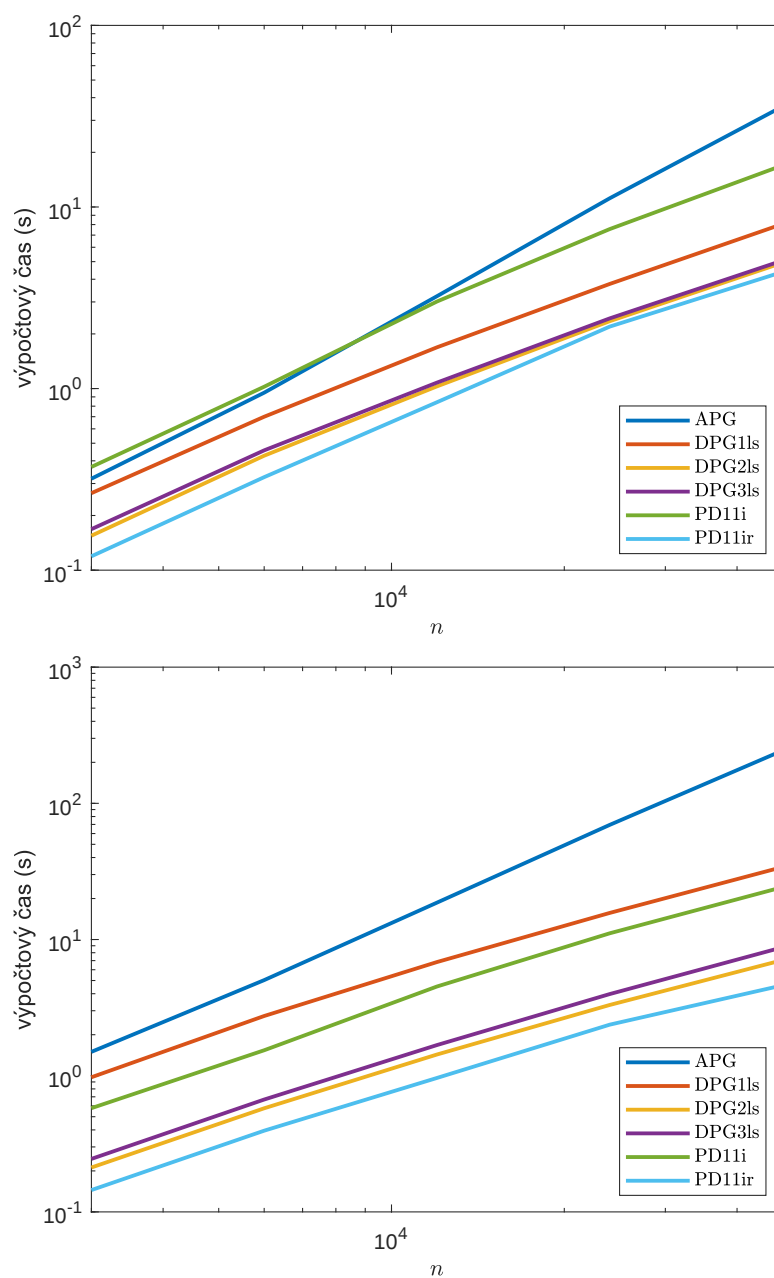
V experimente skúmame vplyv zmeny rozmeru n na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval 19 hodín. Skúmané hodnoty rozmeru n možno nájsť v nasledujúcej tabuľke. Ostatné hodnoty

n	3000	6000	12000	24000	48000
-----	------	------	-------	-------	-------

parametrov generátora úloh ostali fixné na základných hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.76 až B.80 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra n a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Pre relatívnu suboptimalitu riešení 10^{-6} a 10^{-9} možno pozorovať veľký rozdiel medzi PD11i a PD11ir, metóda PD11ir riešenia dosahuje za niekoľkonásobne rýchlejší výpočtový čas a potrebuje menšie množstvo iterácii. Takisto pozorujeme minimálny nárast výpočtového času pre metódu PD11ir pri prechode z riešenia s relatívnou suboptimalitou 10^{-6} na riešenie s relatívnou suboptimalitou 10^{-9} vďaka redukcii dimenzie dát. Z proximal-gradientných metód jednoznačne najlepší výkon podali metódy DPG2ls a DPG3ls. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek pri úlohe (1.2).

Na Obr. 6.22 možno pozorovať vplyv zmeny rozmeru n na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Prirodzene, náročnosť riešených úloh narastá so zväčšovaním rozmeru n , výpočtový čas metód rastie. Takisto pozorujeme, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas metód APG a DPG1ls ako pri ostatných metódach na Obr. 6.22. V experimente celkovo najlepšie výpočtové časy dosahovala metóda PD11ir s redukciou dimenzie a to pre všetky sady úloh pri výpočte riešení s relatívnou suboptimalitou 10^{-6} a 10^{-9} . Kompetitívne výsledky dosiahli aj metódy DPG2ls a DPG3ls najmä pri relatívnej suboptimalite riešení 10^{-6} .



Obr. 6.22: Vplyv rozmeru n na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.7).

6.3.5 Vplyv rozmerov $m \times n$

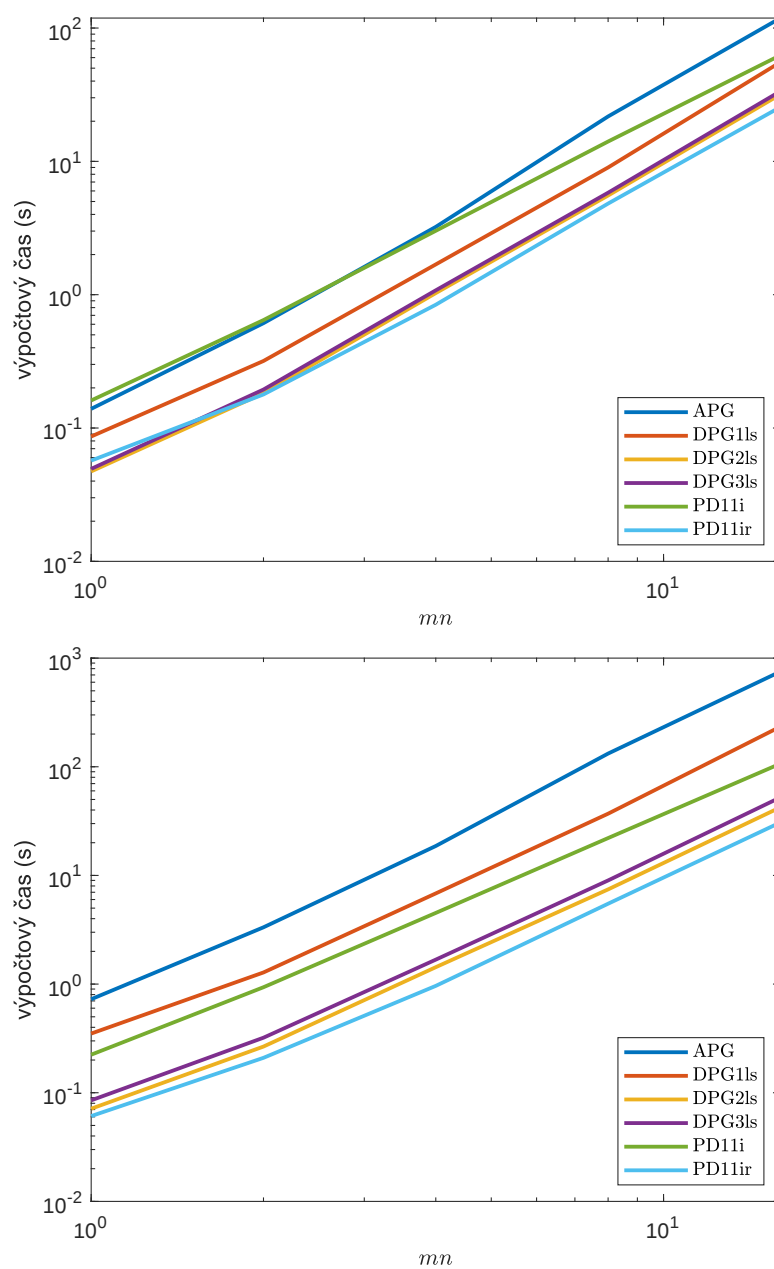
V experimente skúmame vplyv zmeny oboch rozmerov m a n na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 27,5 hodiny. Rozmery $m \times n$ skúmaných úloh (1.7) sa nachádzajú v nasledujúcej tabuľke. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných

m	600	1200	2400	4800	9600
n	3000	6000	12000	24000	48000

hodnotách.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.81 až B.85 z Dodatku B, pričom každá tabuľka prislúcha iným rozmerom skúmaných úloh (1.7) a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Vo finálnej Tabuľke B.85 sme ďalej netestovali proximálne metódy APG, DPG2 a DPG3, je možné pri nich však pozorovať, že z hľadiska počtu iterácii podávali identický výkon až po presnosť 10^{-6} , pričom metóda APG má oveľa menej výpočtovo náročnú iteráciu. Pri relatívnej suboptimálnosti riešení 10^{-9} však metóda APG dosiahla už horší výpočtový čas a potrebovala oveľa viac iterácii. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek. Z tabuliek pri porovnaní metód PD11i a PD11ir pozorujeme úsporu v počte iterácii ako aj vo výpočtovom čase pre presnosti 10^{-6} a 10^{-9} .

Na Obr. 6.23 možno pozorovať vplyv zmeny oboch rozmerov m a n na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Prirodzene, náročnosť riešených úloh narastá so zväčšovaním rozmerov m a n , výpočtové časy metód rastú. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls, pričom PD11ir dosahuje o niečo lepšie výpočtové časy vo všetkých sadách úloh pri presnostiach 10^{-6} a 10^{-9} okrem prvej najjednoduchšej sady úloh pri presnosti 10^{-6} . Výkon PD11ir sa v porovnaní s inými metódami zlepšuje so sprísňujúcou sa presnosťou. Keď metóda PD11ir už dostatočne zredukuje dimenziu dát, presnejšie riešenie dopočíta veľmi rýchlo za pomerne malý čas navyše. Keby sme chceli ešte presnejšie riešenia, rozdiel medzi PD11ir a inými metódami by bol ešte väčší.



Obr. 6.23: Vplyv rozmerov $m \times n$ na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.7).

6.3.6 Vplyv riedkosti matice dát

V nasledujúcom experimente skúmame vplyv riedkosti matice dát, resp. vplyv hustoty zaplnenia de matice dát, na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Výpočet experimentu trval viac ako 37,5 hodín. Skúmame hodnoty parametra de od hodnoty 0.001 až po hodnotu 1, ktorá zodpovedá plne zaplnenej matici dát. Ostatné hodnoty parametrov generátora úloh ostali fixné na základných hodnotách.

Príprava tohto experimentu nám taktiež umožnila pozorovať, pre aké hodnoty parametra de sa oplatí pre porovnávané metódy používať matice typu sparse v prostredí Matlab. Tak ako v prípade úlohy (1.2), väčšina metód má túto hraničnú hodnotu približne na úrovni $de = 0.4$. Pozorovali sme, že pri tejto hodnote je pre výkon metód jemne prospešné maticu dát evidovať v Matlabe už ako plnú. U metód PD13 a PD23 sme pozorovali iné správanie, ich hraničná hodnota bola blízko $de = 0.05$. Pre hodnotu $de = 0.01$ bolo výhodné maticu dát evidovať ako riedku, zatiaľ čo pre hodnotu $de = 0.05$ metóda podávala lepšie výsledky pri matici, ktorú sme evidovali ako plnú. Poznamenajme, že tieto pozorovania sú do veľkej miery ovplyvnené PC zostavou, na ktorej prebiehajú výpočty. Veľký vplyv má napríklad počet jadier procesora.

V experimente vplyvu riedkosti matice dát na výkon metód pri úlohe (1.2) sme zistili, že zmenou parametra de sa pri našom spôsobe generovania dát podstatne menili vlastnosti úloh ako počet nenulových prvkov optimálneho riešenia. Z toho dôvodu uvádzame v nasledujúcej tabuľke priemerný počet nenulových prvkov optimálneho riešenia pre rôzne hodnoty hustoty zaplnenia de pri našom spôsobe generovania úloh (1.7). Vidíme, že pri prvých dvoch sadách úloh máme značne väčšiu priemernú hodnotu nenu-

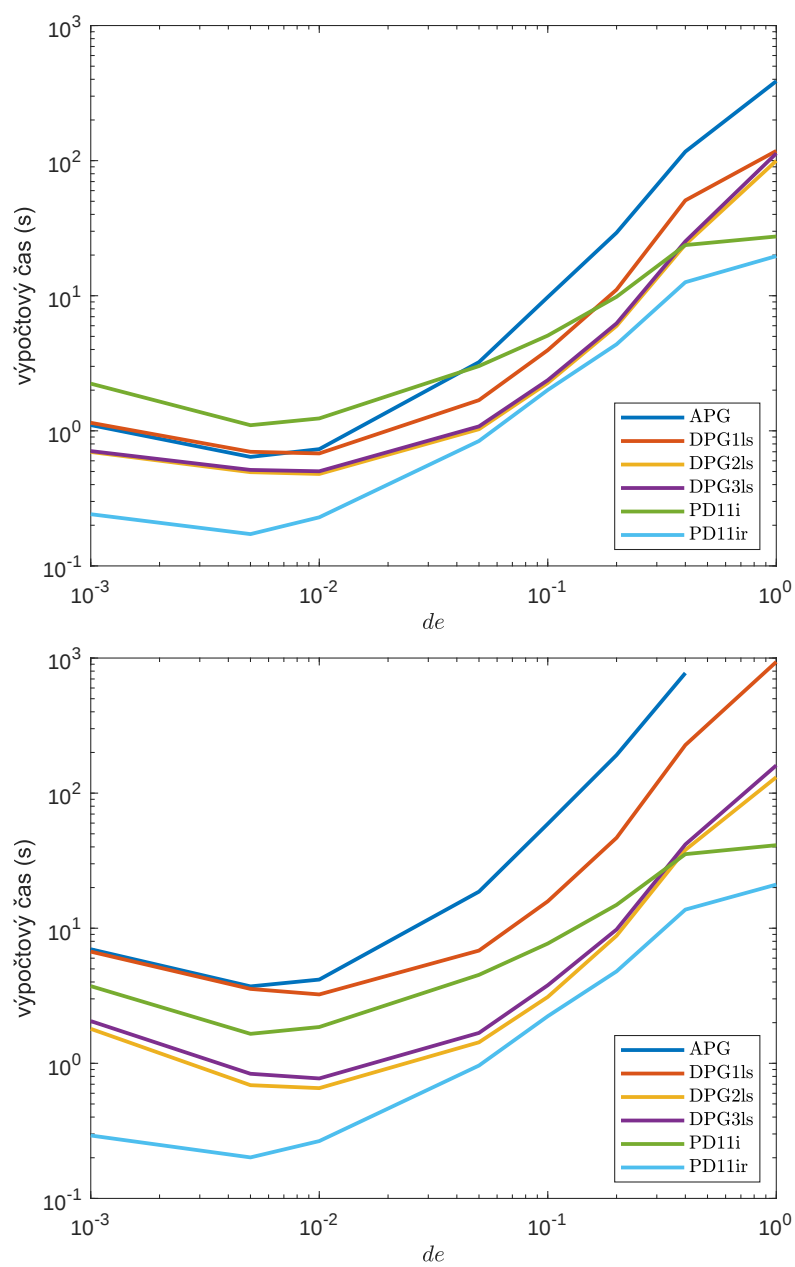
de	0.001	0.005	0.01	0.05	0.1	0.2	0.4	1
$NNZ(x^*)$	1688	1422	1365	1333	1321	1336	1327	1319

lových prvkov optimálneho riešenia. Je preto možné, že prvé dve sady úloh budú ťažšie riešiteľné napriek väčšej riedkosti dátových matíc. Výsledky pre hodnoty $de \geq 0.01$ by mali byť konzistentnejšie a mali mať aj väčšiu výpočtovú hodnotu.

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.86 až B.93 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra de a je zostavená

z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. Ako sme očakávali, väčšina metód v Tabuľke B.86 ($de = 0.001$) má väčší výpočtový čas ako v Tabuľke B.87 ($de = 0.005$). Pre metódu PGls boli niektoré úlohy z prvej sady tak ťažko riešiteľné, že jej nestačil ani výpočtový čas 600s, aby vypočítala riešenie s relatívnou suboptimalitou 10^{-9} , preto sme ju taktisto označili šedou farbou. Pri prechode z Tabuľky B.87 na Tabuľku B.88 niektoré metódy dosiahnú lepší výpočtový čas (napr. DPG1ls, DPG2ls, DPG3ls) a iné naopak horší výpočtový čas (napr. APG, PD11i, PD11ir). Od hodnoty $de \geq 0.01$, t.j. od Tabuľky B.88, sa však metódy správajú viac konzistentne a viac v súlade s očakávaním. V tabuľkách môžeme opäť vidieť mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek. Vidíme, že výpočtové časy mnohých metód prudko rastú so zväčšovaním hustoty zaplnenia de . V Tabuľke B.92 sme z experimentu vyradili aj PDMVB metódy PD13 a PD23. Tieto metódy v niektorých úlohách nevedeli dopočítať riešenie s relatívnou suboptimalitou 10^{-9} , pretože v ich algoritme zlyhalo určovanie dĺžky kroku s . Dĺžka kroku bola znížená na veľmi malú hodnotu a algoritmus stagnoval. Tento jav bol spôsobený tým, že v neskorších fázach algoritmu PD13 a PD23 pracujú s veľmi zle podmienenými maticami.

Na Obr. 6.24 možno pozorovať vplyv zmeny hustoty zaplnenia de matice dát na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Pozorujeme, že náročnosť riešených úloh narastá so zväčšovaním parametra de pre hodnoty $de \geq 0.01$, výpočtové časy metód rastú. Správanie metód pre nižšie hodnoty parametra de bolo do značnej miery ovplyvnené inými „vrodenými“ vlastnosťami generovaných úloh pre prvé dva experimenty. V experimente má jednoznačne najlepší výkon PDMVB metóda PD11ir, ktorá dosiahla najrýchlejšie výpočtové časy pre všetky hodnoty parametra de pri relatívnej suboptimalite riešení 10^{-6} a 10^{-9} . Pre úlohy s plnou maticou ($de = 1$) mala oproti iným metódam veľmi dobrý výpočtový čas aj metóda PD11i.



Obr. 6.24: Vplyv hustoty zaplnenia matice dát de na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.7).

6.3.7 Škálovateľnosť

Na pozorovanie vplyvu veľkosti úloh na výpočtový čas metódy bol v [58] vykonaný test škálovateľnosti. Na nami skúmaných metódach vykonáme preto podobný test.

Generovanie dát

Umelé dáta pre tento experiment generujeme príbuzným spôsobom ako v [58]: Najprv vygenerujeme vektor skupín $b \in \mathbb{R}^m$, ktorý má $0.5m$ zložiek s hodnotou 1 (pozitívny prípad) a $0.5m$ zložiek s hodnotou -1 (negatívny prípad). Následne generujeme náhodnú maticu A s rozmermi $m \times n$, pričom rozmer $m = 0.1n$ a matica má okolo $30m$ nenulových prvkov (hustota zaplnenia $de = 30/n$). Pre pozitívne (negatívne) prípady vytvoríme vektor priemerných hodnôt charakteristík (nenulových) pre danú skupinu p_1 (p_2) so zložkami z rovnomerného rozdelenia na intervale $[0, 1]$ ($[0, -1]$). Charakteristiky pre pozitívne (negatívne) prípady sú nezávislé identicky rozdelené, generované z normálneho rozdelenia $N(0, 1)$ a následne sú posunuté o prislúchajúcu hodnotu z p_1 (p_2).

Optimálne riešenie takto generovanej úlohy nepoznáme. Pri porovnaní metód používame stopovacie kritérium (5.2). Narozdiel od [58] budeme metódy testovať na 50-tich náhodne generovaných úlohách každého typu. Ako štartovací bod volíme $x_0 = 0_n$, $y_0 = 0$.

Kvalifikácia metód

Keďže sa výraznejšie zmenil spôsob generovania dát, spravili sme novú kvalifikáciu metód na úlohe z prostredného experimentu s rozmermi $m = 5000$, $n = 50000$ a regularizačným parametrom $\lambda = 0.1\lambda_{max}$.

Pri primárno-duálnych metódach vnútorného bodu sme použili základné nastavenia (6.1), (6.2) a (6.3), pričom bola pre všetky metódy použitá hodnota $\xi = 0.001$.

Výsledky kvalifikácie sú v Tabuľke 6.4. Metódy, ktoré sú označené šedou farbou, sme sa rozhodli z ďalších experimentov vyradiť. V tabuľke možno pozorovať veľké rozdiely vo výkone metód. Metóda PD11ir kvalifikačnú úlohu vyriešila len za 0.89s a potrebovala na to 21 iterácii. PDMVB metódy PD11, PD21 a prírastkové metódy

PPGd, PPGs, PIAG nedokázali za poskytnutý čas 300s nájsť riešenie ani s relatívnou suboptimálnosťou 10^{-3} . Metódy PD11 a PD21, ktoré riešia Newtonove systémy priamo, nedokázali vypočítať ani jednu iteráciu z dôvodu zaplnenia operačnej pamäte počítača. Možno teda opäť pozorovať veľký prínos PDMVB využívajúcich nepresné smery, metódy PD11i a PD21i úlohu vyriešili bez problémov. Takisto možno pozorovať prínos stochastických algoritmov - stochastická metóda PSAG úlohu vyriešila až do presnosti 10^{-9} , zatiaľ čo deterministická metóda PIAG úlohu nedokázala vyriešiť ani do prvej požadovanej presnosti. Pri metódach MISOd, MISOs, PIAG, PSAG uvádzame najlepší výsledok, ktorý sme dokázali dosiahnuť voľbou dĺžky kroku tvaru $t = 2^k t_0$. Všetky prírastkové metódy však na kvalifikačnej úlohe nepodávali dostatočne dobrý výkon, a preto vyradíme metódy MISOd, MISOs, PPGd, PPGs, PIAG, PSAG a PSAGls. Medzi vyradené metódy patria aj PDMVB metódy PD12, PD13i, PD22, PD23i ako aj proximal-gradientná metóda PG.

Pri porovnaní výsledkov z predošlou kvalifikáciou môžeme pozorovať, že daný spôsob generovania dát viac napomáha proximal-gradientným metódam, mnohé z nich podávajú lepší výkon. Naopak, mnohé PDMVB podávajú horší výkon ako v predošlej kvalifikácii.

Tabuľka 6.4: Kvalifikácia metód pre základnú úlohu (1.7) s $m = 5000$, $n = 50000$, $de = 6E - 04$, $\lambda = 0.1\lambda_{max}$.

Presnosť ϵ	10^{-3}		10^{-6}		10^{-9}		300s	
Metóda	čas	it.	čas	it.	čas	it.	ϵ	it.
APG	0,19	276	1,27	1825	7,46	10721		
APGls	0,42	296	2,73	1982	16,08	11708		
DPG1	0,37	276	2,39	1825	13,69	10499		
DPG1ls	0,25	157	1,46	978	7,58	5131		
DPG2	0,37	276	2,41	1825	4,63	3528		
DPG2ls	0,25	157	0,86	567	1,59	1057		
DPG3	0,36	276	2,38	1825	4,65	3574		
DPG3ls	0,25	157	0,92	608	1,81	1204		
MISOd	158,93	1405680	–	–	–	–	1,6E-04	2652833
MISOs	39,76	351600	–	–	–	–	1,1E-06	2643497
PD11	–	–	–	–	–	–	–	–
PD11i	0,62	14	4,82	20	7,96	27		
PD11ir	0,66	14	0,78	17	0,89	21		
PD12	56,43	15	91,66	21	128,52	27		
PD12i	1,39	15	7,27	21	11,44	28		
PD13	33,32	15	46,63	21	64,04	27		
PD13i	4,95	15	99,89	21	–	–	1,2E-07	29
PD21	–	–	–	–	–	–	–	–
PD21i	1,61	14	11,85	20	19,64	27		
PD22	77,54	14	158,54	20	263,89	27		
PD22i	4,94	15	17,22	21	24,60	27		
PD23	31,25	14	44,65	20	67,14	27		
PD23i	7,96	14	167,40	20	–	–	7,7E-07	23
PG	6,07	8957	47,18	69698	133,00	196445		
PGls	1,56	1836	11,44	13854	31,82	38644		
PPGd	–	–	–	–	–	–	1,8E-02	5468572
PPGs	–	–	–	–	–	–	1,3E-02	5429662
PIAG	–	–	–	–	–	–	2,4E-03	4495490
PSAG	9,09	137040	56,32	836640	150,46	2246640		
PSAGls	27,32	391200	203,54	2886000	–	–	1,5E-07	4234394

Vplyv rozmernosti úloh

V experimente skúmame vplyv rozmernosti úloh na výsledný výpočtový čas a počet iterácii metód pre 3 požadované presnosti (5.3). Ako štartovací bod volíme pre všetky metódy nulový vektor. Výpočet experimentu trval 27 hodín. Skúmame vplyv rozmernosti úloh riadenej parametrom n pre hodnoty z nasledujúcej tabuľky. Ostatné hodnoty

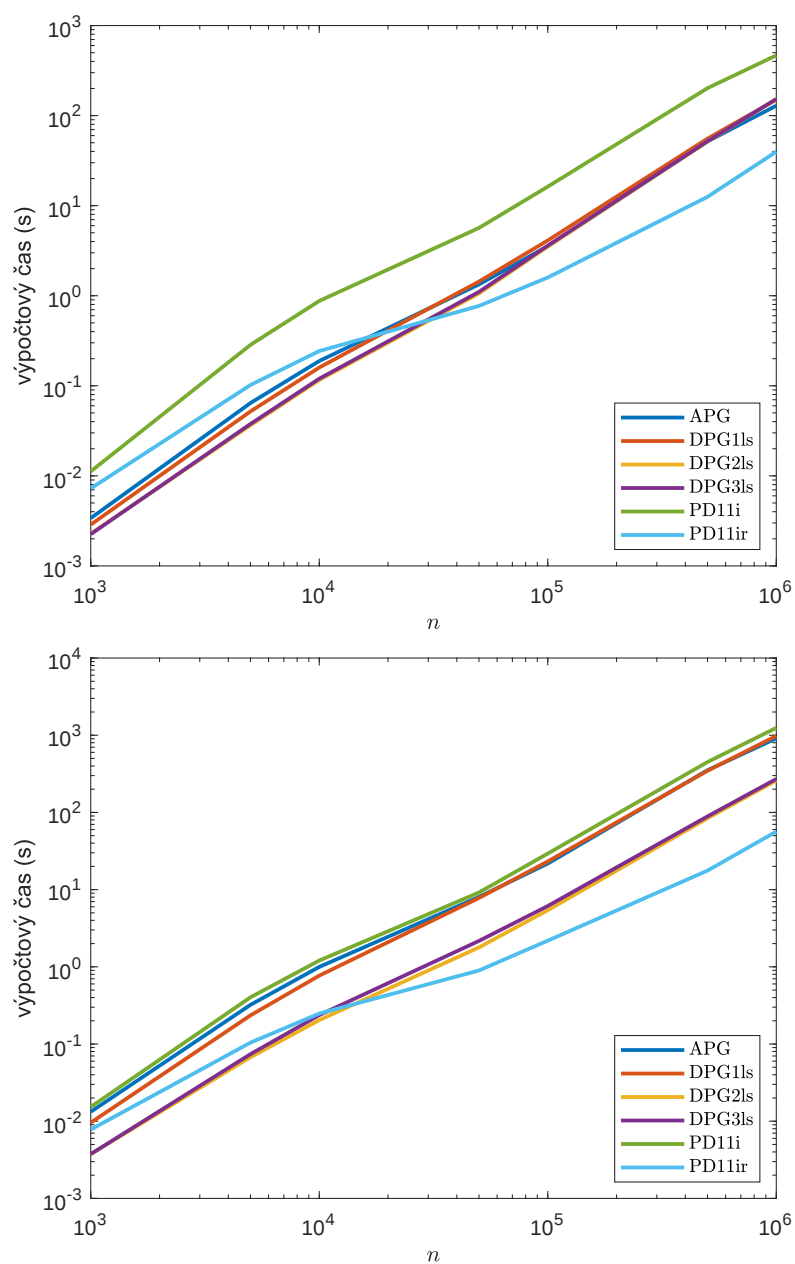
n	1000	5000	10000	50000	100000	500000	1000000
-----	------	------	-------	-------	--------	--------	---------

parametrov generátora úloh sú odvodené z hodnoty rozmeru n .

Podrobné výsledky experimentu sú uvedené v Tabuľkách B.94 až B.100 z Dodatku B, pričom každá tabuľka prislúcha inej hodnote skúmaného parametra n a je zostavená z výpočtov na 50-tich úlohách. Metódy, ktoré sme sa rozhodli pri danom experimente ďalej netestovať sú zvýraznené šedou farbou a uvádzame pri nich výsledok iba z prvej úlohy testovanej sady. V tabuľkách opäť vidíme mnohé pozorovania opísané v časti 6.2.2 o globálnych pozorovaniach z tabuliek. Z Tabuliek môžeme jasne pozorovať, že výpočtové časy metód prudko narastajú s rastúcou rozmernosťou úloh. Medzi metódami sú veľké rozdiely vo výpočtových časoch. V Tabuľkách B.99 a B.100 pre vysokorozmerné úlohy sme preto na všetkých 50-tich úlohách testovali iba tri metódy - PDMVB metódu PD11ir s redukciou dimenzie a proximal-gradientné metódy DPG2ls a DPG3ls. Pri metódach DPG1ls, DPG2ls a DPG3ls pozorujeme veľký prínos modifikovaného algoritmu určovania dĺžky kroku, oproti metódam s konštantnou dĺžkou kroku DPG1, DPG2, DPG3 dosahujú značne rýchlejší výpočtový čas. Pre presnosti 10^{-6} a 10^{-9} možno pozorovať veľký rozdiel medzi PD11i a PD11ir, metóda PD11ir presnosti dosahuje za značne rýchlejší výpočtový čas a potrebuje menšie množstvo iterácii. V Tabuľke B.100 pre úlohy s rozmermi $10^5 \times 10^6$ metóda PD11ir priemerne dosahovala riešenie s relatívnou suboptimálnosťou 10^{-9} len za 57s, zatiaľ čo druhá metóda DPG2ls potrebovala výpočtový čas na úrovni 259s.

Na Obr. 6.25 možno pozorovať vplyv rozmernosti úloh riadenej parametrom n na výpočtový čas pre vybrané metódy, medzi ktorými sú aj metódy podávajúce najlepšie výsledky v experimente. Vidíme, že výpočtový čas všetkých metód prudko rastie so zväčšovaním rozmernosti úloh, náročnosť riešených úloh prudko narastá. Takisto

môžeme pozorovať, že sprísnenie požadovanej presnosti má väčší vplyv na výpočtový čas proximal-gradientných metód APG a DPG1ls ako v prípade ostatných metód na Obr.25. V experimente celkovo najlepšie obstáli metódy PD11ir, DPG2ls, DPG3ls. Pre menej rozmerné úlohy dosahujú najlepšie výpočtové časy DPG2ls a DPG3ls. Pre veľkorozmerné úlohy podáva jednoznačne najlepší výkon metóda PD11ir. Za víťaza testu škálovateľnosti preto považujeme metódu PD11ir.



Obr. 6.25: Vplyv rozmernosti úlohy na výpočtový čas metód pre požadovanú presnosť 10^{-6} a 10^{-9} pri riešení náhodne generovaných úloh (1.7).

Kapitola 7

Aplikácie

V tejto časti otestujeme vybrané metódy na reálnych aplikáciách úloh (1.2) a (1.7) z rôznych oblastí. Konkrétne sa budeme venovať úlohe dizajnu mechanickej konštrukcie, ktorá patrí k úlohám optimálneho topologického dizajnu. Z oblasti spracovania signálu sa budeme venovať úlohe rekonštrukcie riedkeho signálu. A nakoniec metódy otestujeme na úlohe l_1 -regulovanej logistickej regresie aplikovanej na reálne dáta z oblasti medicíny, kategorizácie textu či oblasti detekcie nebezpečných internetových adries a spamu.

Metódy vyberieme na základe výsledkov numerických experimentov uvedených v predošlej kapitole. Keďže primárno-duálne metódy vnútorného bodu pre druhý spôsob transformácie úloh (4.35) podávali v každom experimente horší výkon ako ich ekvivalenty (napr. PD21i vs. PD11i) pre prvý spôsob transformácie (4.34), rozhodli sme sa ich už netestovať. Z PDMVB budeme teda používať iba metódy pre prvú transformáciu úlohy (4.34), t.j. metódy PD11/i, PD12/i, PD13/i a našu PDMVB s redukciou dimenzie PD11ir. Prírastkové metódy, či už deterministické či stochastické (MISOd, MISOs, PPGd, PPGs, PIAG, PSAG, PSAGls), sa nám tiež neosvedčili, preto ich už používať nebudeme. Zo zostávajúcich „nevyradených“ metód budeme pri daných experimentoch používať tie najvhodnejšie, pričom výber zdôvodníme.

Tak ako v predošlej časti o numerických experimentoch budeme sledovať výpočtový čas a počet iterácií, ktoré metódy potrebovali na dosiahnutie riešenia pri troch sledovaných presnostiach. Keďže optimálne riešenia týchto úloh nepoznáme, pre metódy používame stopovacie kritérium (5.2), kde hodnotu duálnej účelovej funkcie získavame z PDMVB.

7.1 Dizajn mechanickej konštrukcie

Jednou z mnohopočetných aplikácií úlohy l_1 -regulovaných najmenších štvorcov (1.2) je aj dizajn rozloženia nosnej konštrukcie (angl. Truss Topology Design) [7], [85], [8], [64], [61], [63]. Pod touto nosnou konštrukciou (angl. truss) rozumieme mechanickú konštrukciu vytvorenú z elastických priečok rôznych objemov, ktoré sú vzájomne medzi sebou pospájané v tzv. uzloch (angl. nodes). Uzly rozlišujeme na dva typy - môžu byť fixné (nepohyblivé) alebo voľné. Štruktúry, akými sú napríklad stĺpy elektrického vedenia, železničné mosty, Eiffelova veža alebo kostry dopravných prostriedkov ako sú lode či lietadlá, možno považovať za takéto mechanické konštrukcie. Fixné uzly možno chápať ako kotviace body, napr. piliere mostov či základy budov. Mechanická konštrukcia môže byť vystavená externej záťaži / silám, pod ktorými sa konštrukcia deformuje (natahovanie priečok, pohyb voľných uzlov). Cieľom dizajnu rozloženia nosnej konštrukcie je pre danú mriežku uzlov a sily pôsobiace na tieto uzly zostrojiť konštrukciu s maximálnou pevnosťou, ktorá je schopná čo najlepšie odolať pôsobiacim silám.

V jednoduchom probléme dizajnu mechanickej konštrukcie máme danú mriežku uzlov rozmeru $r \times c$ (v prípade 2D), z toho $m/2$ uzlov je voľných a ostatné sú fixné. Každý pár z týchto uzlov, ktorý obsahuje aspoň jeden voľný uzol, môže byť prepojený priečkou konštrukcie. Premennou n označujeme počet potenciálnych priečok. Taktiež máme danú 2D externú záťaž $b \in \mathbb{R}^m$, ktorá pôsobí na voľné uzly (každému voľnému uzlu patrí dvojica indexov v b). Naším cieľom je vytvoriť takú nosnú konštrukciu maximálnej pevnosti, ktorá odolá externej záťaži optimálnym spôsobom. Tento problém sa dá naformulovať do tvaru úlohy l_1 -regulovaných najmenších štvorcov (1.2), kde matica $A \in \mathbb{R}^{m \times n}$ prezentuje reakciu nosnej konštrukcie na prítomnosť priečok, jej stĺpec a_i predstavuje vychýlenie uzlov prislúchajúce k priečke i . Absolútna hodnota dizajnovej premennej $x \in \mathbb{R}^n$ zodpovedá váham priečok konštrukcie [7].

Poznamenajme, že prítomnosť l_1 normy (resp. l_1 -regularizácie) v účelovej funkcii spôsobuje, že úloha podáva riedky vektor riešenia x , čo je v súlade s predstavou o optimálnom geometrickom tvare (dizajne) nosnej konštrukcie - väčšina potenciálnych priečok dostane nulovú váhu (nebudú použité), nenulovú váhu majú len tie podstatné priečky. Tento optimálny geometrický tvar sme schopní vykreslovať na pôvodne zadanú

mriežku uzlov.

Rozmernosť úlohy dizajnu mechanickej konštrukcie ($A \in \mathbb{R}^{m \times n}$) s rastúcou veľkosťou mriežky uzlov prudko narastá - hlavne kvôli pribúdajúcemu množstvu potenciálnych priečok (n). V záujme zredukovania ich množstva budeme používať (tak ako v článku [85]) len priečky vzájomne nepretínajúce sa vo viac ako v jednom bode (priečky pretínajúce sa vo viacerých bodoch vieme zoskladať z menších priečok). Ďalšou zaujímavou vlastnosťou matice A pri úlohe dizajnu mechanickej konštrukcie je jej vrodená super-riedkosť, každý jej stĺpec a_i je veľmi riedky, obsahuje maximálne 4 nenulové prvky. Spomínané vlastnosti matice A pre úlohu dizajnu mechanickej konštrukcie môžeme pozorovať v Tabuľke 7.1.

$r \times c$	m	n	# nnz	% nnz
5×5	50	196	712	7.27
10×10	200	3,097	12,046	1.94
20×20	800	48,915	194,178	0.50
30×30	1,800	246,661	983,222	0.22
40×40	3,200	779,035	3,109,978	0.12
50×50	5,000	1,901,881	7,597,822	0.08
60×60	7,200	3,941,015	15,750,018	0.06
100×100	20,000	30,398,795	121,555,778	0.02

Tabuľka 7.1: Závislosť rozmerov (m, n), počtu nenulových prvkov matice A (# nnz) a riedkosti matice A od rozmeru mriežky uzlov ($r \times c$).

7.1.1 Porovnanie metód

V nasledujúcej časti porovnáme vybrané metódy na 5-tich nami vygenerovaných TTD úlohách. Z toho 4 testované úlohy sú mostového typu a jedna predstavuje konštrukciu ramena žeriavu podľa TTD. Súhrn použitých dát a ich vlastnosti možno vidieť v Tabuľke 7.2.

Pre tieto experimenty sme vybrali metódy podľa ich výsledkov z našich numerických experimentov na umelých dátach a takisto na základne predbežných výpočtov. Pri predbežných výpočtoch podávala základná PG metóda a jej verzia s určovaním dĺžky

$r \times c$	typ	m	n	% nnz
6×40	most	472	17473	0,82
9×27	žeriav	468	18014	0,81
7×51	most	706	38270	0,56
9×65	most	1162	103648	0,34
11×83	most	1818	251978	0,22

Tabuľka 7.2: Súhrn vlastností použitých dát pre úlohu dizajnu mechanickej konštrukcie.

korku v každej iterácii PGLs slabý výkon. Takisto metóda PD11 (resp. PD13i) podávala omnoho slabší výkon ako PD11i (resp. PD13). Preto spomínané metódy nezahrnieme do porovnania. Z proximálnych metód sme použili metódy ADMM, APG, APGLs, DPG1, DPG1ls, DPG2, DPG2ls, DPG3, DPG3ls. Z primárno-duálnych metód vnútorného bodu sme použili PD11i, PD12, PD12i, PD13 a PD11ir.

Parameter λ sa pri TTD úlohách zvykne voliť veľmi malý. V nasledujúcich experimentoch používame $\lambda = 0,0001 \approx 7E-06\lambda_{max}$, kde $\lambda_{max} = \|A^T b\|_\infty$. V experimentoch na umelých dátach v predošlej kapitole sme mohli vidieť, že malá hodnota parametra λ má výrazne negatívny vplyv na výkon metód. Pri riešení úloh TTD sa mnohé metódy potýkali s numerickými problémami. Výkon metód budeme porovnávať pri troch presnostiach 10^{-2} , 10^{-4} , 10^{-7} , aby sme boli schopní aj vizuálne rozlíšiť dané riešenia a výsledné mechanické konštrukcie.

Metóda ADMM v experimentoch používala parameter $t = 1/\sqrt{\lambda\sigma_{min}(A^T A)}$, táto voľba sa pri TTD úlohách ukázala ako najlepšia z hodnôt (4.27). U PDMVB s redukciou dimenzie PD11ir sme pri podmienkach redukcie dimenzie (3.14) a (3.15) používali nastavenie parametrov $\epsilon_{p1} = 10^{-4}$ a $\epsilon_{p2} = 0.05$. Inak sú u PDMVB použité bežné nastavenia (6.1), (6.2).

Výsledky vybraných metód na TTD úlohách možno pozorovať v Tabuľkách 7.3, 7.4, 7.5, 7.6 a 7.7, v ktorých sledujeme čas a počet iterácii, ktoré metódy potrebovali na dosiahnutie požadovaných presností. Okrem spomínaných metód možno v týchto tabuľkách nájsť aj metódy s názvom HYB1 a HYB2, ktoré sme špeciálne zostavili na riešenie TTD úloh kombináciou viacerých metód. Z Tabuľky 7.3 možno vidieť, že najrýchlejšia metóda pre počiatočné presnosti je PD13, tá však neskôr zlyháva pri poskytnutí riešenia

s relatívnou suboptimalitou 10^{-7} (za 420s sme dosiahli riešenie s relatívnou suboptimalitou $2,6E-06$ pri 29960 iteráciach). V algoritme PD13 zlyhalo určovanie dĺžky kroku s . Dĺžka kroku bola znížená na veľmi malú hodnotu a algoritmus stagnoval. Tento jav bol spôsobený tým, že v neskorších fázach algoritmu (pri väčšej presnosti) PD13 pracuje s veľmi zle podmienenými maticami. Z tabuliek možno taktiež vidieť, že metóda PD12, ktorá rieši Newtonov systém (4.42) priamo, podáva omnoho lepšie výsledky ako jej verzia PD12i, ktorá používa na riešenie systému iteračnú metódu Minres. Tento jav je neobvyklý a je spôsobený vrodenu super-riedkosťou matíc A pri TTD úlohách. Newtonov systém (4.42) rozmeru $(m+n) \times (m+n)$ maticu A používa v „pôvodnom stave“, a preto má takisto vlastnosť super-riedkosti. Napríklad pri prvej úlohe tento systém obsahuje len 153941 nenulových prvkov, čo predstavuje hustotu zaplnenia matice systému (4.42) menšiu ako 0,05%. Metóda PD12 podáva veľmi dobré výsledky pri výpočte riešenia s relatívnou suboptimalitou 10^{-7} . Na základe týchto výsledkov sme sa rozhodli zostaviť hybridnú metódu HYB1, ktorá v počiatočných iteráciách používa metódu PD13. Následne, ak je pokles duálnej medzery η už malý, potom používame metódu PD12. Týmto spôsobom dorátame riešenie veľkej presnosti za menší čas. Metóda HYB2 predstavuje finálnu podobu našej metódy špecializovanej na TTD úlohy a jedná sa o modifikovanú metódu HYB1, pri ktorej po prvej fáze pomocou PD13 nasleduje fáza s metódou PD12 spolu s redukciou dimenzie (podobne ako pri metóde PD11ir). Hybridná metóda HYB2 preto vo svojom algoritme kombinuje celkovo tri metódy: PD12, PD13 a PG metódu používanú na redukciu dimenzie.

Pre prvé dve TTD úlohy nájdené riešenia s relatívnou suboptimalitou 10^{-2} , 10^{-4} a 10^{-7} vykresľujeme na Obr. 7.1 a Obr. 7.2. Na obrázkoch vykresľujeme len tie priečky, ktoré dostali pri optimalizácii najväčšiu váhu. Týmto spôsobom sme schopní riešenia dosiahnuté pomocou metód aj vizuálne porovnať. Na Obr. 7.1 možno vidieť vykreslené riešenia s relatívnou suboptimalitou 10^{-2} , 10^{-4} a 10^{-7} dosiahnuté pomocou metód ADMM, APG, DPG1ls, DPG2ls, PD11i a PD11ir pri mostovej TTD úlohe s rozmerom 472×17473 . Všetky vykreslené riešenia okrem riešení s relatívnou suboptimalitou 10^{-2} pri metódach PD11i a PD11ir zreteľne pripomínajú mostovú konštrukciu. Zreteľne vidíme 4 piliere mostu (fixné body) a takisto možno pozorovať vozovku vo výške 1, kde boli pri zadávaní úlohy aplikované vertikálne sily, ktorým má konštrukcia odolať.

Obrázky pre presnosť 10^{-4} zreteľne obsahujú menšie množstvo priečok ako v prípade prvej presnosti. Podobne to platí pre presnosť 10^{-7} . Dokonca aj pri tejto presnosti však možno pozorovať jemné rozdiely medzi riešeniami s rovnakou relatívnou suboptimalitou poskytnutými rôznymi metódami. To znamená, že TTD úlohy je potrebné riešiť veľmi presne. Preto je v tabuľkách najpodstatnejší posledný stĺpec pre presnosť 10^{-7} .

Na Obr. 7.2 možno vidieť vykreslené riešenia s relatívnou suboptimalitou 10^{-2} , 10^{-4} a 10^{-7} dosiahnuté pomocou metód ADMM, APG, DPG1ls, DPG2ls, PD11i a PD11ir pri TTD úlohe žeriavového typu s rozmerom 468×18014 . Z vykreslených riešení je možné pozorovať konštrukciu pripomínajúcu rameno žeriavu. Na ľavo je konštrukcia ukotvená vo fixných bodoch. Táto konštrukcia má následne odolávať vertikálnym silám, ktoré sú aplikované v pravej časti obrázka vo výške 4. Pri riešeníach s relatívnou suboptimalitou 10^{-2} konštrukcie obsahujú veľa prebytočných priečok. Tak ako v prípade mostových konštrukcií, aj v tomto prípade je možné pozorovať ubúdanie priečok pri postupnom spresňovaní riešenia. Zaujímavé je, že výsledné mechanické konštrukcie pre riešenia s relatívnou suboptimalitou 10^{-7} nevyužívajú všetky fixné body, využívajú prevažne dva krajné fixné body. Finálne konštrukcie sú pomerne porovnateľné, len ťažko sa hľadajú rozdiely.

Ukažku tvarov mostových konštrukcií pre ďalšie úlohy, na ktorých metódy porovnávame, možno nájsť na Obr. 7.5, kde vykresľujeme mostové konštrukcie získané z veľmi presného riešenia daných úloh pomocou PDMVB.

Z Tabuliek 7.3, 7.4, 7.5, 7.6 si možno všimnúť, že metóda ADMM podáva veľmi dobrý výkon hlavne pre počiatkové presnosti. Pri poslednej presnosti sa pri stúpajúcich rozmeroch úloh výrazne zhoršuje. Z Obr 7.1 a 7.2 je však evidentné, že úlohy TTD je potrebné riešiť do veľkej presnosti. Jednoznačne najlepšie výsledky pri TTD úlohách dosahuje hybridná metóda HYB2, potom nasledujú metódy HYB1 a PD12. Mimo týchto metód, ktoré veľmi efektívne využívajú riedkosť matice A , najlepšie obstáli metódy PD11ir, PD11i a APG (pre menšie rozmery aj metóda ADMM). Z tabuliek je takisto možné postrehnúť, že metódy DPG2/ls, DPG3/ls fungujú na dané úlohy výrazne horšie ako metódy APG, DPG1/ls, čo sme pozorovali v tejto práci málokedy. Tento jav by mohol byť spôsobený veľmi slabým výkonom PG metódy (ktorú tieto metódy používajú) na týchto úlohách. Spomínali sme, že mnohé metódy mali pri počítaní

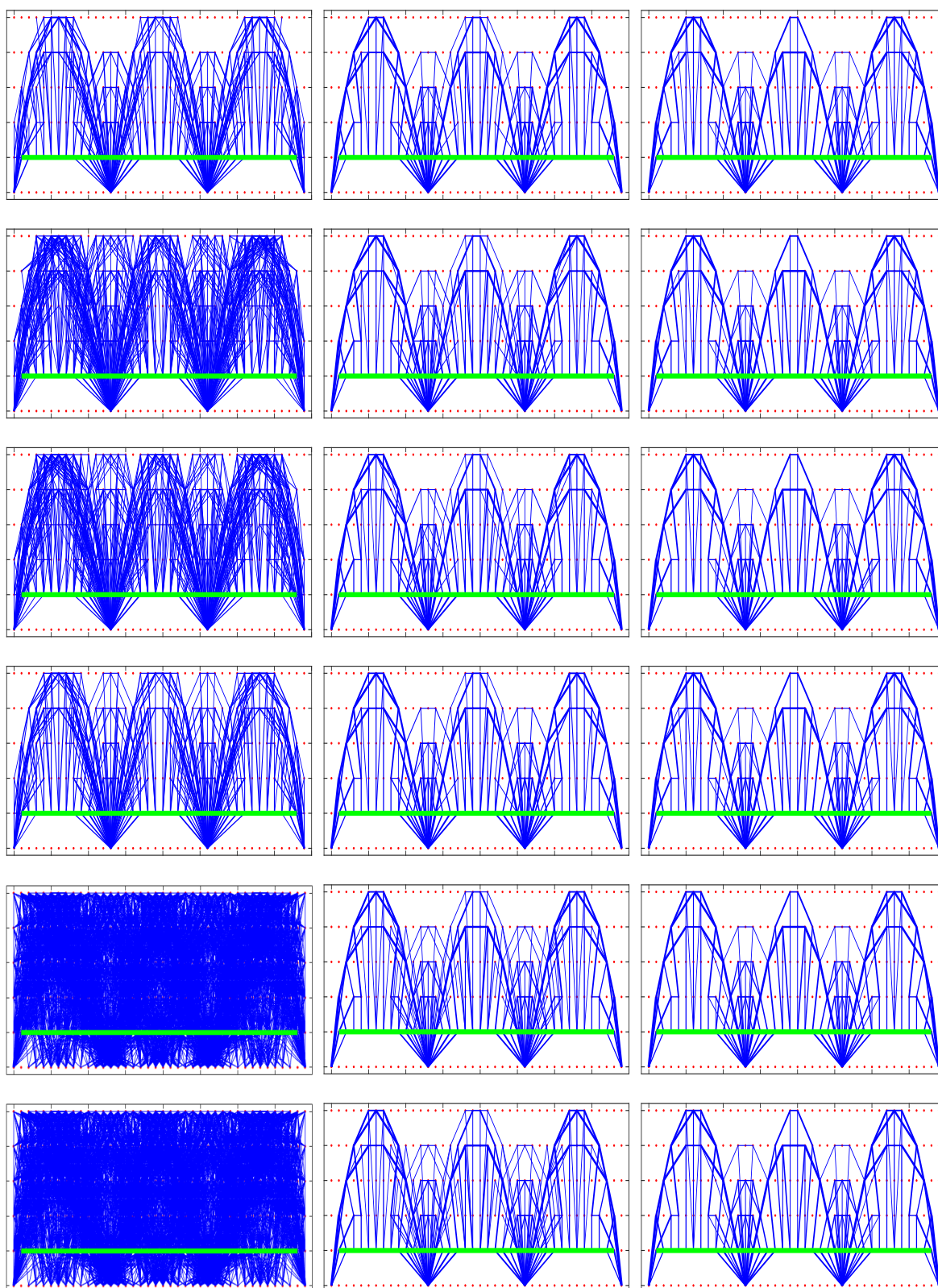
úloh TTD numerické problémy. Pri metódach DPG2ls a DPG3ls pôvodne na 2 TTD úlohách zlyhalo určovanie dĺžky kroku z dôvodu porovnávania veľmi podobných čísel. Dĺžka kroku bola následne zmenšená na príliš malú hodnotu a metódy následne už konvergovali príliš pomaly. Toto zlyhanie nastalo pri úlohách z Tabuliek 7.4 a 7.5, kde metódy pôvodne nevedeli dosiahnuť riešenie s relatívnou suboptimalitou 10^{-7} . Metódy sa nám pre tieto úlohy podarilo opraviť pridaním dodatočných podmienok.

V Tabuľkách 7.3, 7.4, 7.5, 7.6 takisto môžeme pozorovať, že metóda PD11ir má spočiatku podobný výkon ako PD11i no akonáhle začne redukovať dimenziu začne šetriť výpočtový čas. Tento jav je pozorovateľný aj na Obr. 7.3, kde pre TTD úlohu s rozmerom 1162×103648 porovnávame vývoj relatívnej suboptimality v čase pre metódy ADMM, DPG1, DPG1ls, PD11i, PD11ir. Na tomto obrázku takisto môžeme vidieť, že metódy ADMM a APG nie sú spádové. Na Obr. 7.4 porovnávame pre tú istú úlohu vývoj relatívnej suboptimality v čase pre metódy PD12, HYB1 a HYB2. Z obrázku možno pozorovať ako veľmi metóde HYB1 pomohol štart metódou PD13 pri porovnaní s pôvodnou metódou PD12, a takisto vidíme ako metóde HYB2 pomáha redukcia dimenzie oproti metóde HYB1. Podobné pozorovania možno urobiť aj napríklad z Tabuľky 7.7.

V Tabuľke 7.7 možno nájsť porovnanie najúspešnejších metód PD12, HYB1 a HYB2 na finálnej TTD úlohe najväčších rozmerov. Jedná sa o úlohu mostového typu na mriežke 11×83 s viac ako 250000 prípustnými priečkami.

Tabuľka 7.3: Porovnanie metód pre mostovú TTD úlohu s $m = 472$, $n = 17473$.

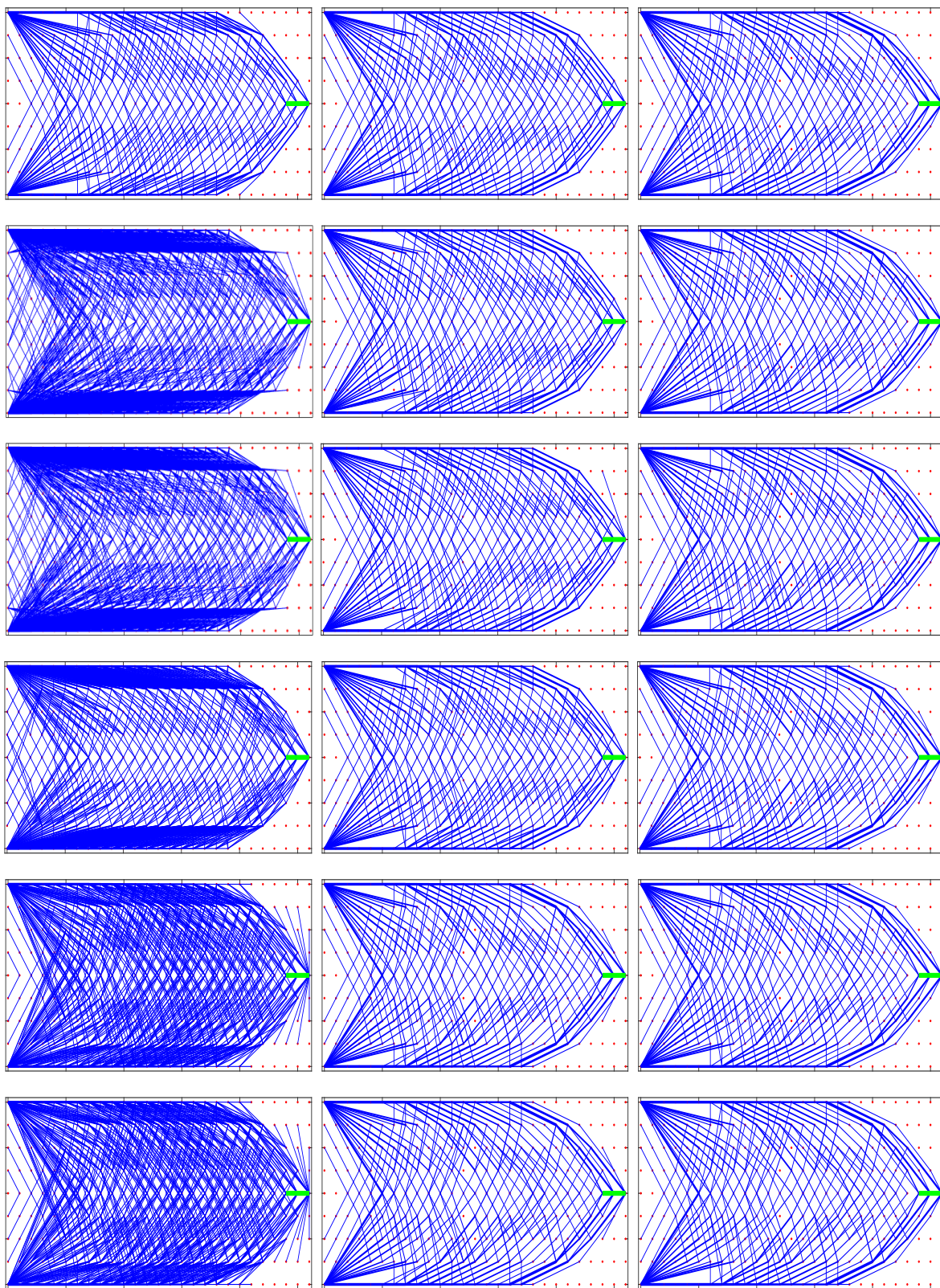
Presnosť ϵ	10^{-2}		10^{-4}		10^{-7}		$420s$	
Metóda	čas	it.	čas	it.	čas	it.	ϵ	it.
ADMM	0,25	789	4,26	14567	20,44	69850		
APG	2,14	17466	11,68	93725	21,15	170251		
APGls	5,13	20399	27,99	109388	58,25	225794		
DPG1	3,64	17411	19,11	95012	33,23	168514		
DPG1ls	3,47	12652	16,11	58741	26,37	94785		
DPG2	6,04	30404	51,02	259782	82,31	417664		
DPG2ls	6,04	21010	38,84	133925	64,12	220668		
DPG3	8,86	42449	82,80	416150	135,80	684392		
DPG3ls	7,12	24466	66,23	227938	109,82	377066		
PD11i	6,26	18	13,54	22	18,96	32		
PD11ir	6,27	18	13,63	22	16,26	30		
PD12	0,83	19	1,06	24	1,53	33		
PD12i	15,28	20	41,50	28	57,11	37		
PD13	0,14	19	0,17	24	–	–	2,6E-06	29960
HYB1	0,13	19	0,16	24	0,56	34		
HYB2	0,13	19	0,16	24	0,26	32		



Obr. 7.1: Porovnanie riešení ADMM, APG, DPG1ls, DPG2ls, PD11i, PD11ir pre presnosti 10^{-2} , 10^{-4} , 10^{-7} pri TTD úlohe mostového typu s rozmermi 472×17473 .

Tabuľka 7.4: Porovnanie metód pre „žeriavovú“ TTD úlohu s $m = 468$, $n = 18014$.

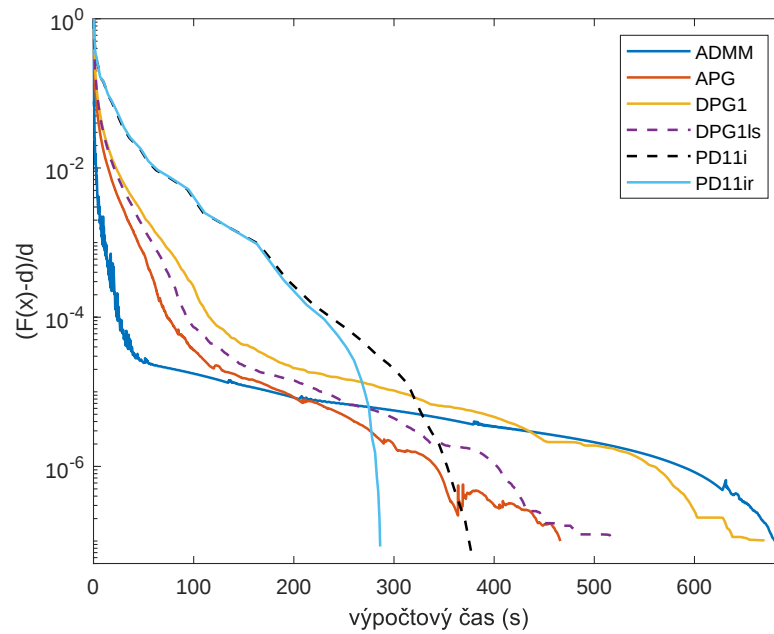
Presnosť ϵ	10^{-2}		10^{-4}		10^{-7}	
Metóda	čas	it.	čas	it.	čas	it.
ADMM	0,24	766	1,83	5834	36,12	114805
APG	2,18	17604	11,25	87294	66,90	524240
APGls	5,17	20701	25,73	101924	204,07	765238
DPG1	3,72	17619	18,08	87434	137,90	688775
DPG1ls	3,50	12577	14,59	52868	98,97	350032
DPG2	6,27	31248	69,33	344675	232,75	1153597
DPG2ls	5,56	19121	59,34	211626	216,00	783218
DPG3	8,34	40093	114,58	564375	409,45	2026587
DPG3ls	6,92	23709	104,65	377755	397,72	1449409
PD11i	11,41	27	40,78	35	66,38	43
PD11ir	12,36	27	42,54	35	59,49	42
PD12	1,75	28	2,21	35	2,71	42
PD12i	15,06	26	96,70	36	150,98	43
PD13	0,19	28	0,24	35	1,73	147
HYB1	0,18	28	0,23	35	0,46	46
HYB2	0,19	28	0,24	35	0,33	45



Obr. 7.2: Porovnanie riešení ADMM, APG, DPG1ls, DPG2ls, PD11i, PD11ir pre presnosti 10^{-2} , 10^{-4} , 10^{-7} pri TTD úlohe žeriavového typu s rozmermi 468×18014 .

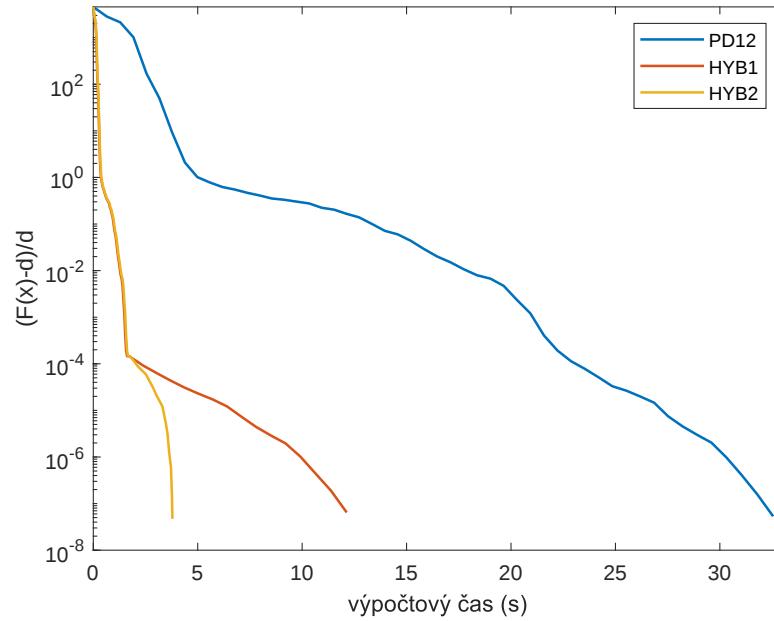
Tabuľka 7.5: Porovnanie metód pre mostovú TTD úlohu s $m = 706$, $n = 38720$.

Metóda	10^{-2}		10^{-4}		10^{-7}	
	čas	it.	čas	it.	čas	it.
ADMM	0,72	1068	20,55	31319	624,20	960412
APG	5,24	18736	42,90	156323	183,15	663589
APGls	10,71	22095	88,27	184550	372,39	777060
DPG1	7,95	18719	66,29	155566	308,24	725275
DPG1ls	7,00	13348	48,97	93942	208,35	396685
DPG2	11,96	27993	188,01	437768	600,62	1400512
DPG2ls	9,97	19022	149,94	301875	496,90	1014258
DPG3	14,71	34285	348,31	809095	1094,49	2542114
DPG3ls	9,98	19432	273,47	555858	914,44	1876744
PD11i	29,31	23	98,66	33	133,91	44
PD11ir	28,16	23	87,25	32	107,33	41
PD12	3,03	22	4,01	28	5,91	39
PD12i	30,53	26	347,91	47	449,95	59
PD13	0,33	22	0,42	28	897,28	27018
HYB1	0,33	22	0,43	28	1,70	44
HYB2	0,34	22	0,44	28	0,67	41

Obr. 7.3: Porovnanie vývoja relatívnej suboptimality v čase pre metódy ADMM, APG, DPG1, DPG1ls, PD11i, PD11ir pri TTD úlohe rozmerov 1162×103648 .

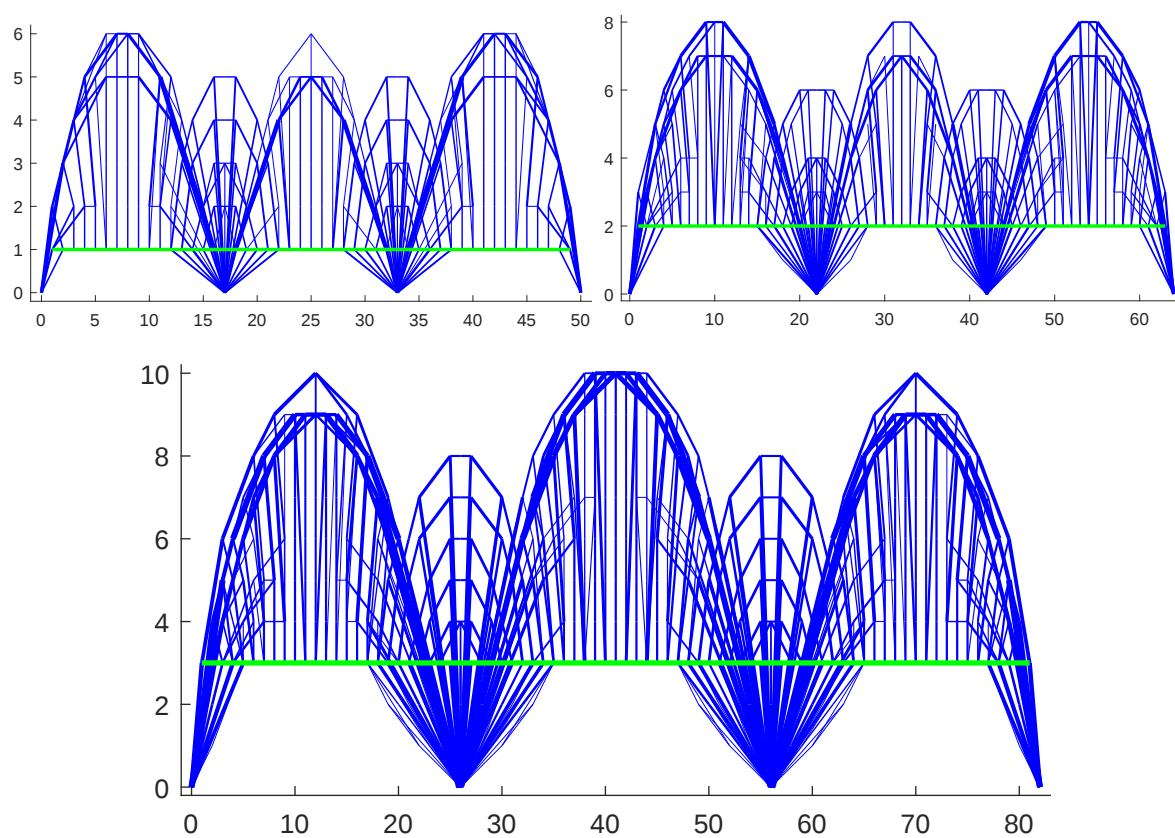
Tabuľka 7.6: Porovnanie metód pre mostovú TTD úlohu s $m = 1162$, $n = 103648$.

Presnosť ϵ	10^{-2}		10^{-4}		10^{-7}	
Metóda	čas	it.	čas	it.	čas	it.
ADMM	2,29	1114	23,63	11692	680,06	338310
APG	14,65	20861	73,62	109160	465,99	690615
APGls	31,85	24654	166,54	129395	1057,71	817875
DPG1	22,50	20889	115,68	107527	669,66	623499
DPG1ls	20,50	14383	93,22	65540	517,19	363570
DPG2	27,55	25718	464,12	430786	1410,63	1310219
DPG2ls	26,27	18425	329,11	229630	981,42	685147
DPG3	29,72	27732	743,65	690315	2421,98	2248242
DPG3ls	31,16	21571	526,15	367703	1720,32	1199625
PD11i	62,43	30	239,15	38	377,85	52
PD11ir	64,18	30	230,03	38	286,21	50
PD12	18,38	30	23,53	38	32,55	51
PD12i	241,87	46	1100,39	72	1657,71	87
PD13	1,31	30	928,86	9083	4547,55	44372
HYB1	1,31	30	2,39	39	12,13	53
HYB2	1,35	30	2,14	39	3,79	51

Obr. 7.4: Porovnanie vývoja relatívnej suboptimality v čase pre metódy PD12, HYB1, HYB2 pri TTD úlohe rozmerov 1162×103648 .

Tabuľka 7.7: Porovnanie metód pre mostovú TTD úlohu s $m = 1818$, $n = 251978$.

Metóda	Presnosť ϵ		10^{-2}		10^{-4}		10^{-7}	
	čas	it.	čas	it.	čas	it.	čas	it.
PD12	67,88	36	91,03	48	126,35	65		
HYB1	4,86	36	8,74	50	42,18	66		
HYB2	4,94	36	7,99	50	14,17	64		

Obr. 7.5: Ukážka tvaru mostov pre TTD úlohu s rozmermi 706×38270 , 1162×103648 , 1818×251978 .

7.2 Rekonštrukcia riedkeho signálu

Medzi aplikácie úlohy l_1 -regulovaných najmenších štvorcov (1.2) patrí aj tzv. rekonštrukcia riedkeho signálu (angl. Sparse Signal Reconstruction). Tzv. compressed sensing problémy získali v poslednej dobe dosť popularity v mnohých oblastiach - napríklad v strojovom učení, štatistike, medicíne či v oblasti spracovania signálov. Jedným z najvýznamnejších predstaviteľom týchto problémov je úloha rekonštrukcie riedkeho signálu (angl. sparse signal reconstruction (SSR)), ktorej základný tvar vyzerá nasledovne:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \|x\|_0 \\ \text{s.t. } Ax = b, \end{aligned}$$

kde $b \in \mathbb{R}^m$ je vektor pozorovaní, $x \in \mathbb{R}^n$ je vektor neznámych (vektor rekonštruovaného signálu) a $A \in \mathbb{R}^{m \times n}$ je dátová matica, ktorá predstavuje merania. Pre rozmery matice A platí $m < n$, t.j. počet pozorovaní je menší ako počet neznámych. Keďže l_0 norma je rovná počtu nenulových zložiek vektora, na úlohu rekonštrukcie riedkeho signálu možno pozeráť ako na hľadanie najredšieho riešenia lineárneho systému $Ax = b$. Úloha rekonštrukcie riedkeho signálu je vďaka prítomnosti l_0 normy NP ťažká, vyžaduje kombinatorickú optimalizáciu. Ukázalo sa však, že za určitých podmienok [19], [29] je možné nájsť presné riešenie tejto úlohy riešením lepšie riešiteľnej úlohy

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \|x\|_1 \\ \text{s.t. } Ax = b. \end{aligned}$$

V praxi však vektor pozorovaní b často obsahuje aj šum, a preto nemá zmysel hľadať presné riešenie systému $Ax = b$. V tom prípade sa riešenie s riedkou štruktúrou dá hľadať riešením nami skúmanej úlohy l_1 -regulovaných najmenších štvorcov (1.2), pričom parameter $\lambda > 0$ kontroluje riedkosť nájdeného riešenia.

7.2.1 Porovnanie metód

V nasledujúcej časti porovnáme vybrané metódy na 3 úlohách rekonštrukcie riedkeho signálu.

Pre tieto experimenty sme vybrali metódy podľa ich výsledkov z našich numerických experimentov na umelých dátach a takisto na základne predbežných výpočtov. Vlast-

nosti matice meraní A týchto úloh (matice sú veľmi dobre podmienené) nadržiavajú proximálnym metódam prvého rádu, preto sme použili metódy ADMM, APG, APGLs, DPG1, DPG1ls, DPG2, DPG2ls, DPG3, DPG3ls, PG a PGLs. Z primárno-duálnych metód vnútorného bodu sme použili PD11i, PD12i, PD13 a PD11ir. Metódy PD11 a PD12, ktoré riešia Newtonov systém priamo, podávali omnoho horší výkon ako ich ekvivalenty využívajúce iteračné metódy, a preto ich v tabuľkách neuvádzame. Výkon metód budeme porovnávať pri troch presnostiach 10^{-3} , 10^{-6} , 10^{-9} .

Metóda ADMM v experimentoch používala parameter $t = 1/\sqrt{\lambda\sigma_{\min}(A^T A)} = 1/\sqrt{\lambda}$, táto voľba sa pri SSR úlohách ukázala ako najlepšia z hodnôt (4.27). U PDMVB využívajúcich iteračné metódy (PD11i, PD12i, PD11ir) používame bežné nastavenia parametrov (6.1), (6.2), (6.3) okrem parametra $\xi = 0.005$, ktorý riadi ako presne iteračnými metódami riešime Newtonove systémy.

Ako prvú úlohu uvažujme úlohu rekonštrukcie riedkeho signálu, kde originálny riedky signál $\tilde{x} \in \mathbb{R}^{4096}$ obsahuje 160 náhodne rozložených nenulových prvkov s hodnotami $\{-1, 1\}$. Tento originálny riedky signál možno vidieť na Obr. 7.6. Matica meraní A (reprezentujúca 1024 pozorovaní signálu) je vytvorená najprv vygenerovaním náhodnej matice rozmerov 1024×4096 s nezávislými identicky rozdelenými prvkami z normálneho rozdelenia $N(0, 1)$. Následne sme riadky matice zortonormalizovali podobne ako v [21], [56]. Takto vytvorená matica má málo nulových prvkov. Vektor pozorovaní $b \in \mathbb{R}^{1024}$ vytvoríme ako

$$b = A\tilde{x} + e,$$

kde vektor $e \in \mathbb{R}^m$ predstavuje šum v pozorovaní a jeho prvky sú z normálneho rozdelenia $N(0, 10^{-4})$. Hodnota regularizačného parametra λ bola nastavená ako

$$\lambda = 0.01\lambda_{\max} = 0.01\|A^T b\|_{\infty},$$

čo je bežne používaná hodnota pri úlohách rekonštrukcie riedkeho signálu.

Druhú úlohu sme vytvorili analogickým spôsobom ako prvú, akurát s vyššími rozmermi. Konkrétne, matica meraní A má rozmery 8000×32000 a originálny riedky signál $\tilde{x} \in \mathbb{R}^{32000}$ obsahuje 1250 náhodne rozložených nenulových prvkov s hodnotami $\{-1, 1\}$.

V prípade tretej úlohy sa jedná o náš nápad ako inak reprezentovať signál, keďže

vykreslovať signál (predošlého typu) veľkých rozmerov by bolo nepraktické. Ako signál možno chápať aj napríklad zvuk, obrázky či video. Je známy fakt, že obrázky majú pri niektorých transformáciách riedku štruktúru ako napríklad pri vlnkovej transformácii. Náš prístup je o niečo jednoduchší. Použijeme čierne biely obrázok vo veľkosti 256×256 , z ktorého spravíme negatív a následne ho vektorizujeme. Tým získame originálny riedky signál $\tilde{x} \in \mathbb{R}^{65536}$, ktorý má 2646 nenulových prvkov s hodnotou 1. Tento originálny riedky signál, resp. pôvodný čiernobiely obrázok, možno vidieť na Obr. 7.9. Ako maticu meraní $A \in \mathbb{R}^{16384 \times 65536}$ sme zvolili náhodnú podmaticu ortogonálnej DCT matice rozmeru 65536×65536 . Výsledná matica A , ktorá je plná, zaberala v operačnej pamati počítača 8GB. Vektor pozorovaní b sme vytvorili analogickým spôsobom ako v predošlých úlohách.

Zhrnutie používaných dát a ich vlastnosti možno nájsť v Tabuľke 7.8.

m	n	# nen. prvkov $\tilde{x}$	hodnoty nen. prv. $\tilde{x}$	RAM
1024	4096	160	$\{-1,1\}$	32 MB
8000	32000	1250	$\{-1,1\}$	1.91 GB
16384	65536	2646	1	8 GB

Tabuľka 7.8: Súhrn vlastností použitých dát pri úlohe rekonštrukcie riedkeho signálu.

Výsledky vybraných metód na SSR úlohách možno pozorovať v Tabuľkách 7.9, 7.10 a 7.11, v ktorých sledujeme čas a počet iterácií, ktoré metódy potrebovali na dosiahnutie požadovaných presností. V týchto tabuľkách takisto uvádzame tzv. mean squared error dosiahnutých riešení, t.j.

$$MSE := \frac{1}{n} \sum_{i=1}^n (x_i - \tilde{x}_i)^2. \quad (7.1)$$

Okrem spomínaných metód je v týchto tabuľkách prítomná aj metóda s názvom ADMMi. Jedná sa o ADMM metódu, ktorej pri Choleského rozklade matice $I + tAA^T$ napovieme, že matica $AA^T = I$. Pri compressed sensing problémoch sú však riadky matíc často takmer ortonormálne [38], preto je pre porovnanie metód relevantný aj riadok s pôvodnou metódou ADMM.

Z Tabuliek 7.9., 7.10, 7.11 možno pozorovať, že proximálne metódy ADMM, APG/l_s, DPG1/l_s, DPG2/l_s, DPG3/l_s a PG/l_s dané SSR úlohy riešia rádovo za stovky iterácií.

Najviac iterácii (1009) potrebovala PG metóda na nájdenie riešenia s relatívnou suboptimalitou 10^{-9} pri prvej úlohe. Toto pozorovanie je v silnom kontraste s predošlými TTD úlohami, kde tieto metódy potrebovali väčšinou stovky tisíc iterácii (niektoré aj viac ako 2milióny). Tento jav je spôsobený aj veľmi dobrými spektrálnymi vlastnosťami matíc meraní A . Vďaka tomu, že sú matice dobre podmienené, dané metódy fungujú veľmi dobre. Z testovaných metód celkovo najlepší výkon podávajú metódy DPG2ls, DPG3ls a ADMMi, väčšinou v tomto poradí. Takisto si možno všimnúť, že PD11ir ako jediná metóda z PDMVB drží s týmito metódami prvého rádu krok, hlavne pri výpočte riešenia s relatívnou suboptimalitou 10^{-9} . Ďalej možno pozorovať, že metóda ADMMi vykonáva identické množstvo iterácii ako ADMM, ale časový rozdiel medzi týmito metódami nenastáva len na začiatku metód pri vykonaní choleského rozkladu. To nám napovedá, že metóda ADMM v iteráciách nepracovala s tak presným (a riedkym) rozkladom matice $I + tAA^T$ ako metóda ADMMi. Z Tabuliek 7.9., 7.10, 7.11 možno pri PD11ir vidieť, že metóda má spočiatku podobný výkon ako PD11i, no akonáhle začne redukovať dimenziu začne šetriť výpočtový čas a iterácie. Pri finálnej presnosti 10^{-9} tak PD11ir všetky SSR úlohy vyriešila za menej ako tretinový čas a potrebovala vždy o 5 iterácii menej ako pôvodná PD11i metóda. Tento jav je pozorovateľný aj na Obr. 7.8, kde pre SSR úlohu s rozmermi 16384×65536 porovnávame vývoj relatívnej suboptimality v čase pre metódy ADMM, ADMMi, APG, DPG2ls, DPG3ls, PD11i a PD11ir. Na tomto obrázku takisto môžeme vidieť typické „skoky“ APG metódy a zároveň časový interval, v ktorom metóda ADMM vykonávala Choleského rozklad.

Z hodnôt MSE v Tabuľkách 7.9., 7.10, 7.11 možno usúdiť, že naozaj rekonštruujeme originálny signál, pretože finálne hodnoty MSE sú pre všetky testované metódy menšie ako $2E - 04$. Avšak môžeme pozorovať zaujímavý jav, že podľa tejto odchýlky boli niektoré metódy k pôvodnému signálu bližšie s riešením s relatívnou suboptimalitou 10^{-3} ako s riešením s relatívnou suboptimalitou 10^{-9} , kde už všetky metódy dosahujú rovnakú hodnotu MSE. Najvýraznejšie je tento jav prítomný pri metóde DPG1ls. Najhoršiu hodnotu MSE pri počiatočnej presnosti dosahovala PG metóda.

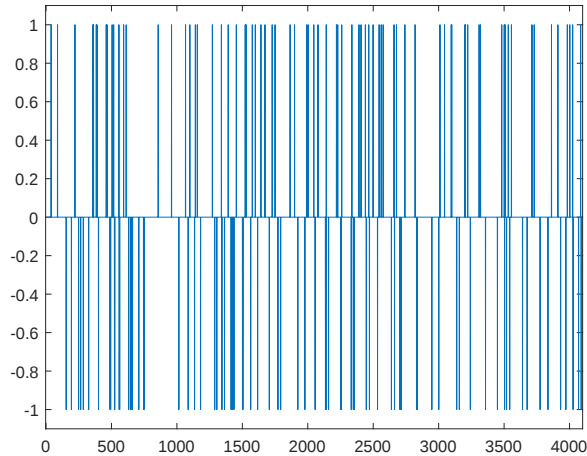
Na Obr. 7.7 preto vykresľujeme výsledné rekonštrukcie riedkeho signálu pomocou metód PG, DPG1ls a PD11i pre presnosti 10^{-3} a 10^{-9} pre SSR úlohu rozmeru 1024×4096 . Originálny signál je vykreslený na Obr. 7.6. Z Obr. 7.7 vidieť, že výsledné

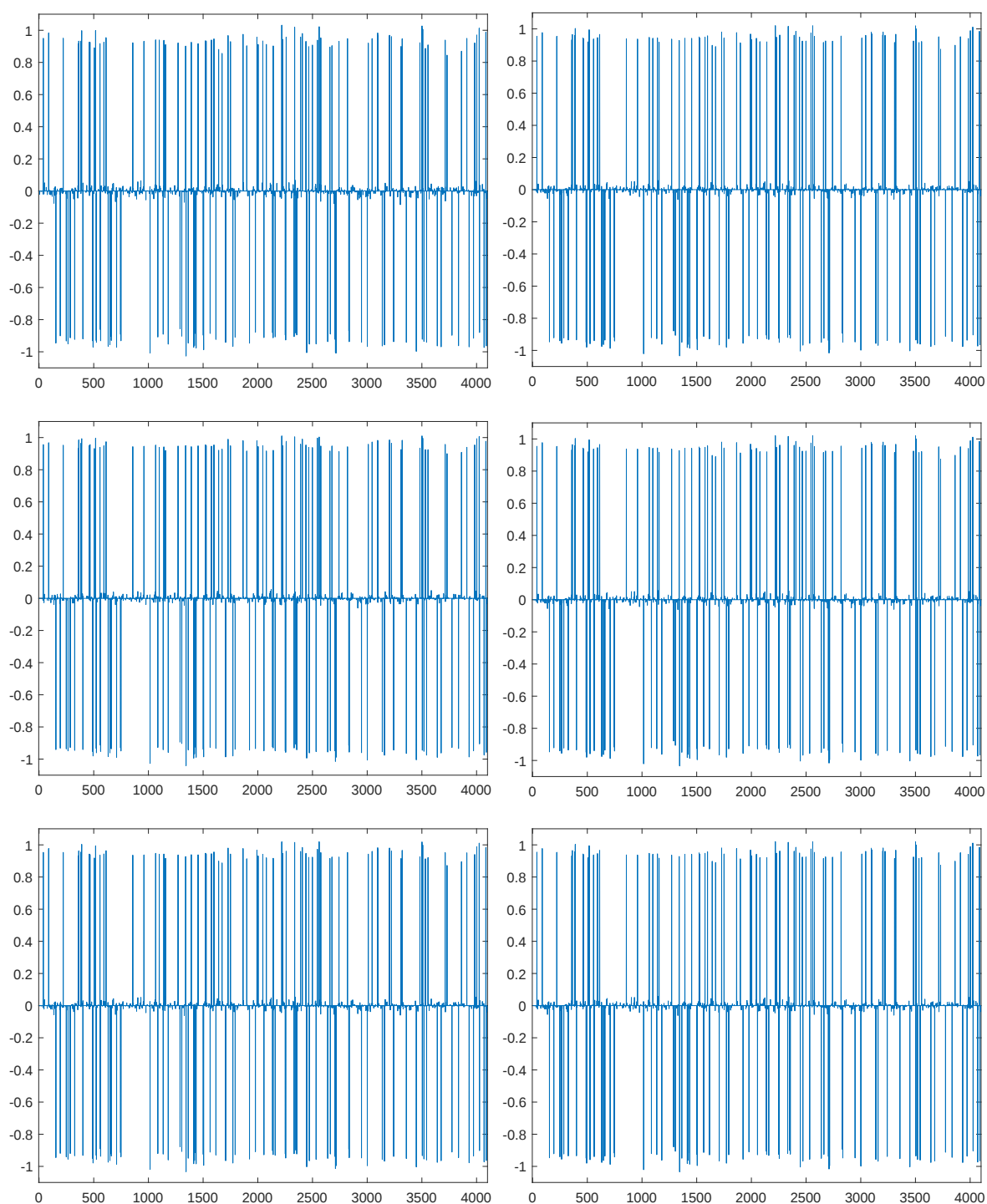
rekonštrukcie presne našli všetky pozície nenulových prvkov pôvodného signálu z oveľa menej pozorovaní ako je dĺžka signálu (resp. počet neznámych). Ak vieme, že pôvodný signál mal nenulové prvky s hodnotami $\{-1, 1\}$, tak by sme z výsledných riešení vedeli originálny signál zrekonštruovať úplne presne. Takisto môžeme z obrázkov pozorovať, že sa výsledné riešenia vizuálne len ťažko porovnávajú, a to platí pri porovnaní medzi metódami, ale aj medzi presnosťami 10^{-3} a 10^{-9} . Možno pozorovať len veľmi jemné rozdiely, napr. pre prvú presnosť pri metóde DPG1ls (toto riešenie malo najmenšiu MSE zo všetkých riešení) možno pozorovať, že niektoré nenulové hodnoty sú bližšie k hranici $\{-1, 1\}$ ako u ostatných riešení. Aj keď má riešenie pomocou PD11i všetky prvky nenulové tento jav nie je možné pozorovať na obrázku pri porovnaní s ostatnými metódami, ktoré poskytli riešenia s riedkou štruktúrou.

K podobným záverom možno dospieť aj pri pozorovaní Obr. 7.10, kde vykresľujeme výsledné rekonštrukcie riedkeho signálu pomocou metód PG, DPG1ls a PD11i pre presnosti 10^{-3} , 10^{-6} a 10^{-9} pre SSR úlohu rozmeru 16384×65536 . Vidíme, že sme zrekonštruovali obrázky na prvý pohľad nerozpoznatelné od seba či od pôvodného signálu (resp. obrázku) zobrazeného na Obr. 7.9. Výsledné rekonštrukcie presne našli všetky pozície nenulových prvkov pôvodného signálu z oveľa menej pozorovaní ako je dĺžka signálu. Riešenia pomocou metód PG, DPG1ls a PD11i sa pri rôznych presnostiach líšia len jemne odlišnou štruktúrou šumu, pričom najviac šumu obsahuje obrázok pre metódu PG pri prvej presnosti (čo naznačuje aj najvyššie dosiahnuté MSE spomedzi testovaných metód). Ostatné riešenia sú len ťažko odlíšiteľné na obrazovke počítača.

Tabuľka 7.9: Porovnanie metód pre SSR úlohu s $m = 1024$, $n = 4096$.

Presnosť ϵ	10^{-3}			10^{-6}			10^{-9}		
	čas	it.	MSE	čas	it.	MSE	čas	it.	MSE
ADMM	0,21	48	1,61E-04	0,35	103	1,87E-04	0,53	175	1,86E-04
ADMMi	0,07	48	1,61E-04	0,15	103	1,87E-04	0,25	175	1,86E-04
APG	0,10	75	1,56E-04	0,26	192	1,86E-04	0,59	442	1,86E-04
APGls	0,16	75	1,56E-04	0,39	192	1,86E-04	0,88	442	1,86E-04
DPG1	0,16	79	1,55E-04	0,40	199	1,86E-04	0,92	463	1,86E-04
DPG1ls	0,13	57	1,48E-04	0,28	126	1,86E-04	0,55	260	1,86E-04
DPG2	0,13	64	2,07E-04	0,31	153	1,86E-04	0,40	196	1,86E-04
DPG2ls	0,11	48	1,91E-04	0,22	99	1,85E-04	0,28	124	1,86E-04
DPG3	0,14	66	2,05E-04	0,22	108	1,85E-04	0,33	160	1,86E-04
DPG3ls	0,11	49	1,90E-04	0,16	72	1,85E-04	0,23	103	1,86E-04
PD11i	0,24	14	1,89E-04	0,57	21	1,86E-04	0,94	27	1,86E-04
PD11ir	0,27	14	1,89E-04	0,29	17	1,86E-04	0,31	22	1,86E-04
PD12i	0,63	14	1,88E-04	1,22	20	1,86E-04	2,04	27	1,86E-04
PD13	0,78	14	1,89E-04	1,15	21	1,86E-04	1,47	27	1,86E-04
PG	0,66	511	2,85E-04	0,95	731	1,88E-04	1,36	1009	1,86E-04
PGls	0,27	182	2,82E-04	0,36	248	1,88E-04	0,48	328	1,86E-04

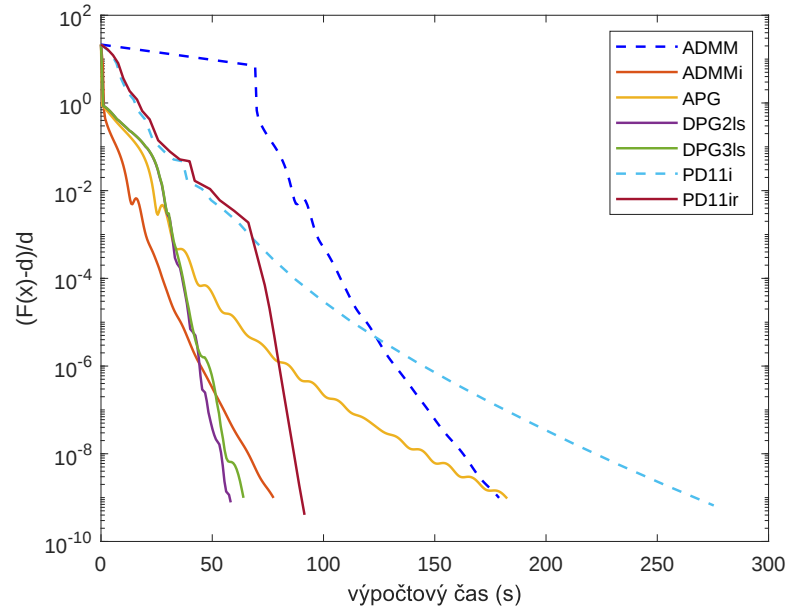
Obr. 7.6: Originálny riedky signál rozmeru $n = 4096$ s 160 náhodne rozloženými nenulovými prvkami s hodnotami $\{-1, 1\}$.



Obr. 7.7: Výsledná rekonštrukcia riedkeho signálu pomocou PG (1. riadok), DPG1ls (2. riadok) a PD1li (3. riadok) metódy pre presnosti 10^{-3} , 10^{-9} .

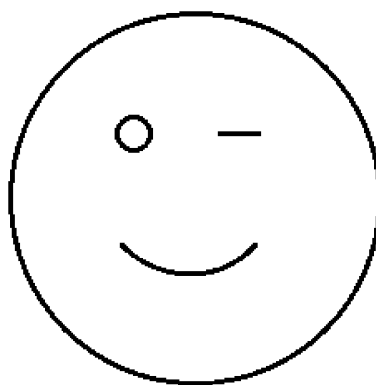
Tabuľka 7.10: Porovnanie metód pre SSR úlohu s $m = 8000$, $n = 32000$.

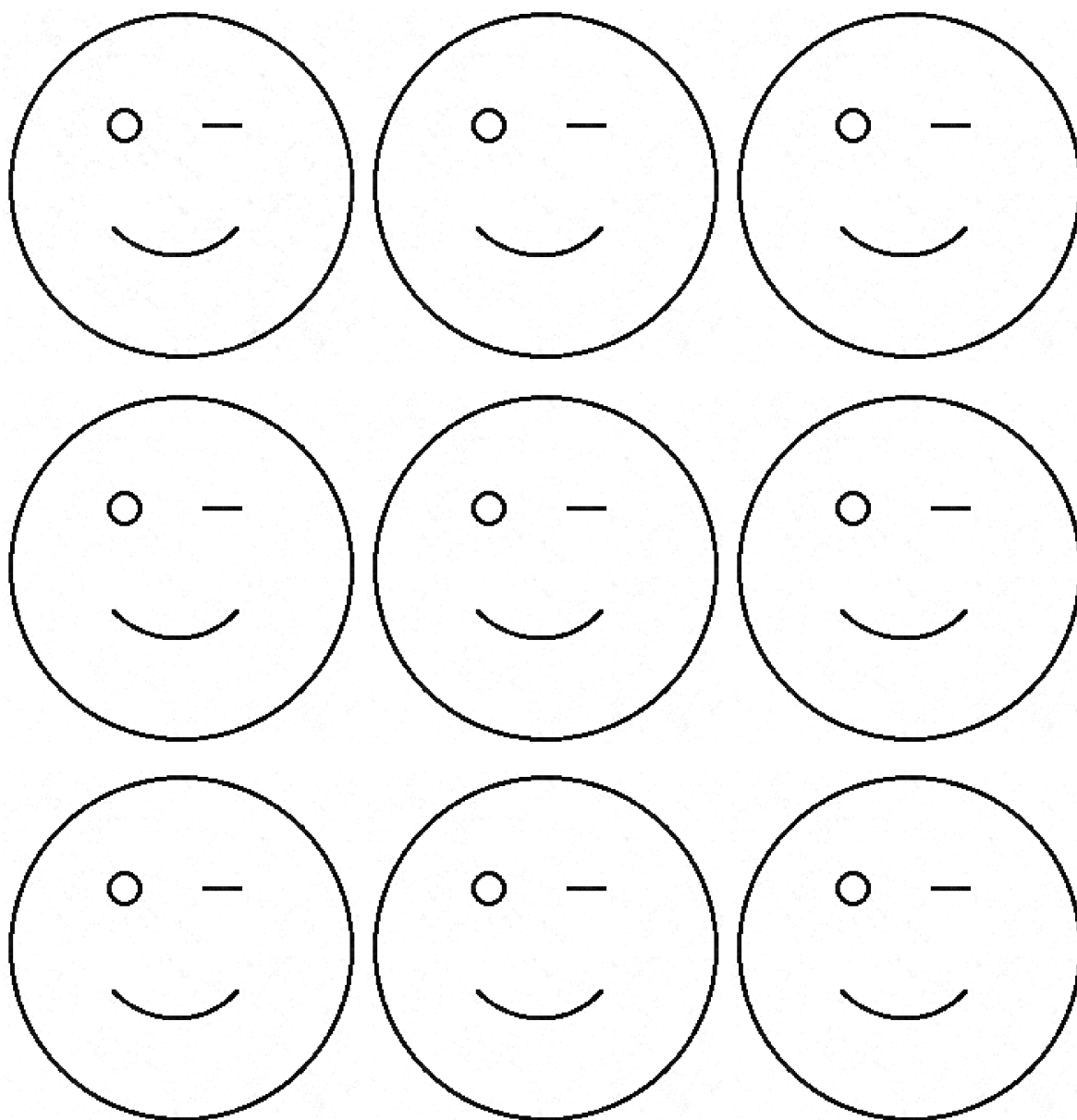
Presnosť ϵ	10^{-3}			10^{-6}			10^{-9}		
Metóda	čas	it.	MSE	čas	it.	MSE	čas	it.	MSE
ADMM	16,66	44	1,60E-04	25,67	96	1,82E-04	37,28	163	1,81E-04
ADMMi	5,47	44	1,60E-04	11,87	96	1,82E-04	20,03	163	1,81E-04
APG	8,25	68	1,55E-04	21,87	181	1,81E-04	48,00	397	1,81E-04
APGls	12,38	68	1,55E-04	33,03	181	1,81E-04	72,41	397	1,81E-04
DPG1	13,04	71	1,62E-04	32,98	180	1,82E-04	72,56	397	1,81E-04
DPG1ls	9,66	51	1,46E-04	21,39	113	1,81E-04	40,52	214	1,81E-04
DPG2	10,81	58	1,64E-04	15,58	82	1,81E-04	20,75	109	1,81E-04
DPG2ls	8,29	43	1,78E-04	11,75	60	1,81E-04	14,84	75	1,81E-04
DPG3	11,13	60	1,63E-04	17,33	94	1,81E-04	37,06	202	1,81E-04
DPG3ls	8,39	44	1,78E-04	12,45	65	1,81E-04	18,30	96	1,81E-04
PD11i	16,56	16	1,85E-04	40,49	23	1,81E-04	71,66	29	1,81E-04
PD11ir	17,65	16	1,85E-04	20,00	20	1,81E-04	22,56	24	1,81E-04
PD12i	35,87	17	1,83E-04	70,90	23	1,81E-04	131,52	30	1,81E-04
PG	51,12	423	2,74E-04	71,48	592	1,83E-04	94,08	779	1,81E-04
PGls	19,21	149	2,67E-04	25,64	198	1,83E-04	32,39	250	1,81E-04

Obr. 7.8: Porovnanie vývoja relatívnej suboptimality v čase pre metódy ADMM, ADMMi, APG, DPG2ls, DPG3ls, PD11i, PD11ir pre SSR úlohu rozmerov 16384×65536 .

Tabuľka 7.11: Porovnanie metód pre SSR úlohu s $m = 16384$, $n = 65536$.

Presnosť ϵ	10^{-3}			10^{-6}			10^{-9}		
	čas	it.	MSE	čas	it.	MSE	čas	it.	MSE
ADMM	97,07	39	1,34E-04	132,19	87	1,46E-04	178,88	151	1,46E-04
ADMMi	20,26	39	1,34E-04	44,82	87	1,46E-04	77,55	151	1,46E-04
APG	31,64	62	1,29E-04	85,17	167	1,46E-04	182,53	358	1,46E-04
APGls	47,47	62	1,29E-04	127,88	167	1,46E-04	274,11	358	1,46E-04
DPG1	50,31	65	1,36E-04	121,91	158	1,46E-04	273,59	355	1,46E-04
DPG1ls	37,56	47	1,14E-04	82,51	103	1,45E-04	151,86	190	1,46E-04
DPG2	41,61	53	1,35E-04	94,06	121	1,46E-04	216,54	280	1,46E-04
DPG2ls	32,44	40	1,59E-04	44,71	54	1,46E-04	58,30	69	1,46E-04
DPG3	42,62	55	1,34E-04	66,49	86	1,46E-04	139,71	181	1,46E-04
DPG3ls	32,77	41	1,58E-04	48,93	61	1,45E-04	64,04	80	1,46E-04
PD11i	68,37	18	1,49E-04	157,82	25	1,46E-04	275,38	31	1,46E-04
PD11ir	73,60	18	1,49E-04	82,18	22	1,46E-04	91,48	26	1,46E-04
PD12i	139,71	17	1,48E-04	268,76	23	1,46E-04	503,85	30	1,46E-04
PG	179,03	351	2,26E-04	252,07	494	1,47E-04	336,25	659	1,46E-04
PGls	66,86	122	2,24E-04	89,26	163	1,47E-04	115,52	210	1,46E-04

Obr. 7.9: Originálny obrázok, ktorého negatív prezentuje originálny riedky signál roz-
meru $n = 65536$ s 2646 nenulovými prvkami s hodnotou 1.



Obr. 7.10: Výsledná rekonštrukcia riedkeho signálu pomocou PG (1. riadok), DPG1ls (2. riadok) a PD11i (3. riadok) metódy pre presnosti 10^{-3} , 10^{-6} , 10^{-9} .

7.3 l_1 -regulovaná logistická regresia

Úlohe logistickej regresie a úlohe l_1 -regulovanej logistickej regresie (1.7) sme sa venovali v prvej kapitole tejto práce. Táto úloha má ako klasifikačný nástroj široké uplatnenie v praxi v mnohých oblastiach. V tejto časti l_1 -regulovanú logistickú regresiu aplikujeme na reálne dáta pochádzajúce z medicíny, z oblasti kategorizácie textu a z oblasti detekovania škodlivých internetových stránok a spamu.

Pri porovnaní metód používame 6 verejne prístupných dátových setov. Dátové sety sme prevažne čerpali zo stránky LIBSVM [24]. Súhrn používaných dát a ich vlastnosti možno vidieť v Tabuľke 7.12. Pôvodné dáta Colon pochádzajú z [1] a dáta Leukemia pochádzajú z [43], v oboch prípadoch sa jedná o dáta ohľadom rakoviny. Dáta Rcv1 sú pôvodom z publikácie [69] a dáta News sú z [54], pričom obidva tieto dátové sety sa týkajú kategorizácie textu. Dáta Url pochádzajú z [74] a dáta Webspam pochádzajú z [103], v oboch prípadoch sa jedná o dáta z oblasti detekovania škodlivých internetových stránok a spamu.

Tabuľka 7.12: Súhrn použitých (trénovacích) dát pre l_1 -regulovanú logistickú regresiu.

Názov	m	n	% nnz	RAM	test. vzorka
Colon	62	2000	100	1 MB	nie
Leukemia	38	7129	100	2,1 MB	áno
Rcv1	20242	47236	0,16	23,2 MB	áno
Url	60000	3231949	0,0036	131,1 MB	áno
News	19996	1355191	0,034	149,2 MB	nie
Webspam	200000	680715	0,55	11,1 GB	áno

Dáta sú spracované nasledujúcim spôsobom. Dáta Colon a Leukemia (plná matica dát) boli štandardizované, aby mali nulový priemer a varianciu 1 pre každé pozorovanie, a následne štandardizované, aby mali nulový priemer a varianciu 1 pre každú charakteristiku [24]. Dáta News a Webspam boli normalizované po pozorovaniach, t.j. každé pozorovanie má jednotkovú dĺžku [24]. Dátové sety Leukemia a Rcv1 obsahujú aj tzv. testovaciu vzorku, aby sme mohli otestovať úspešnosť klasifikácie pomocou l_1 -regulovanej logistickej regresie (nášho modelu). Keďže úspešnosť klasifikácie je zaujímavou otázkou, rozhodli sme sa, že pre dátové sety Url a Webspam si testovaciu

vzorku vytvoríme. Pri dátovom sete Url sme použili dáta z dní 5, 6, 7 ako trénovaciu vzorku a dáta z prvého dňa sme použili ako testovaciu vzorku. V prípade dátového setu Webspam sme z dát vyhodili nulové stĺpce (charakteristiky) a následne sme pozorovania náhodne preusporiadali. Potom sme za trénovaciu vzorku zvolili prvých 200000 pozorovaní a za testovaciu vzorku sme použili zvyšok dát.

7.3.1 Voľba hodnoty regularizačného parametra λ

Regularizačný parameter λ má pri úlohe l_1 -regulovanej logistickej regresie (1.7) veľmi významnú rolu. V experimentoch na umelých dátach sme videli, že hodnota tohto parametra významne ovplyvňuje výkon použitých metód. Jeho hodnota takisto ovplyvňuje počet nenulových zložiek riešenia, t.j. reguluje výber podstatných charakteristík pre klasifikáciu. Zvyčajne platí, že s rastúcou hodnotou λ klesá počet nenulových zložiek riešenia (vektora váh x) [97], [58]. Keďže parameter λ riadi počet nenulových zložiek riešenia, má vplyv aj na samotnú úspešnosť klasifikácie. Keďže chceme metódy otestovať za reprezentatívnych podmienok a relevantnú hodnotu parametra λ pre klasifikáciu pomocou l_1 -regulovanej logistickej regresie, rozhodli sme sa najprv vykonať na dátach analýzu vplyvu parametra λ .

Na Obrázkoch 7.11 - 7.16 možno vidieť vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia. Graf počtu nenulových prvkov riešenia je získaný použitím PDMVB na trénovaciu vzorku dátových setov pre rôzne hodnoty parametra λ . Úlohy boli riešené do veľkej presnosti a následne bolo použité zaokrúhľovacie pravidlo vychádzajúce z podmienok optimality úlohy (1.7), podobne ako v článku [58]. Zložku vektora x_i považujeme za nulovú, ak platí

$$|\nabla_{x_i} f(x, y)| \leq 0.9999\lambda, \quad (7.2)$$

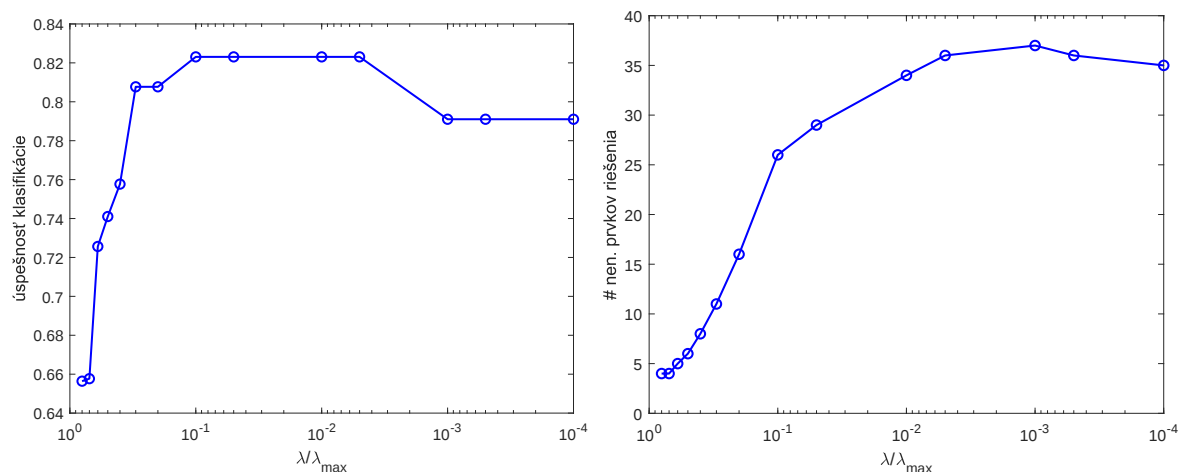
kde $\nabla_x f(x, y)$ má tvar (4.11). Graf úspešnosti klasifikácie získavame otestovaním riešenia (pre danú hodnotu λ) nájdeného pomocou PDMVB na testovacej vzorke dát (pre dátové sety Leukemia, Rcv1, Url a Webspam). Pre pridelovanie tried $\{-1, 1\}$ používame lineárny klasifikátor (1.6) z Kapitoly 1. Pri dátach, ktoré nemajú testovaciu vzorku (dátové sety Colon a News), je potrebné postupovať inak. Na získanie úspešnosti modelu bez testovacej vzorky sa v praxi často používa tzv. k -fold cross-validation. Pri k -fold

cross-validation sa dáta rozdelia na k náhodných rovnako veľkých vzoriek. Z týchto k vzoriek sa jedna použije ako testovacia vzorka a zvyšných $k - 1$ vzoriek sa použije ako trénovacia vzorka. Tento proces krížovej validácie opakujeme k krát s každou jednou z k vzoriek ako testovacou vzorkou. Z týchto k výsledkov úspešnosti modelu následne urobíme priemer, čo predstavuje náš výsledný odhad úspešnosti modelu pre danú hodnotu parametra λ . Týmto spôsobom sú všetky pozorovania používané aj na trénovanie aj na testovanie a každé pozorovanie je na testovanie použité presne jedenkrát. Graf úspešnosti klasifikácie pre dáta Colon je získaný pomocou 5-fold cross-validácie a pre dáta News pomocou 4-fold cross-validácie. Vďaka tejto technike získavame pre dáta bez testovacej vzorky lepší a robustnejší odhad fungovania modelu.

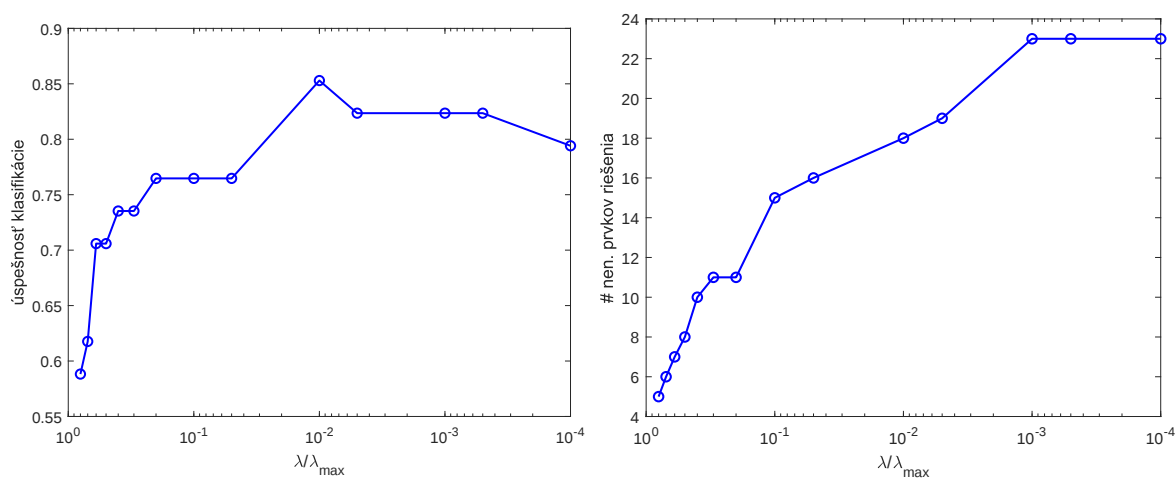
Na Obrázkoch 7.11 - 7.16 možno vidieť vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dátové sady Colon, Leukemia, Rcv1, Url, News a Webspam. Hodnoty regularizačného parametra λ volíme tvaru $\lambda = c\lambda_{max}$, kde hraničná hodnota λ_{max} je určená ako v (1.8). Z obrázkov možno pozorovať, že pre klesajúcu hodnotu parametra λ zvyčajne stúpa počet nenulových zložiek riešenia. Jemný pokles počtu nenulových zložiek sme zaznamenali len pri dátovom sete Colon (Obr. 7.11) pre hodnoty $\lambda \leq 10^{-3}$. Z obrázkov je takisto možné vidieť, že pre klesajúcu hodnotu parametra λ nemusí úspešnosť klasifikácie nutne rásť. l_1 -regulovaná logistická regresia nám teda umožní nielen vybrať podstatné charakteristiky a vďaka tomu získať častokrát lepšie interpretovateľný model, ale aj následná klasifikácia môže často podávať lepšie výsledky (keďže neklasifikujeme pomocou nesignifikantných charakterisík). Tento jav nastáva u dátových setov Colon, Leukemia, Rcv1, Url (u posledných dvoch je však na grafoch ťažko pozorovateľný).

Pri dátových setoch Colon a Leukemia bol výber hodnoty parametra λ jednoduchý. Pre dáta Colon sme zvolili hodnotu $\lambda = 0.1\lambda_{max}$, pri ktorej dosahujeme maximálnu úspešnosť klasifikácie (nad 82% - hodnota pochádza z 5-fold cross-validácie) a riešenie má 26 nenulových prvkov. V prípade dát Leukemia volíme hodnotu $\lambda = 0.01\lambda_{max}$, pri ktorej dosahujeme na testovacej vzorke maximálnu úspešnosť klasifikácie (nad 85%) a riešenie má 18 nenulových prvkov. V prípade dát Rcv1 sa dá za rozumné hodnoty parametra λ považovať hodnoty z intervalu $[10^{-2}\lambda_{max}, 10^{-3}\lambda_{max}]$. Maximálna úspešnosť klasifikácie 95.69% sa dosahuje pre $\lambda = 0.001\lambda_{max}$, pričom dané riešenie má okolo 1680

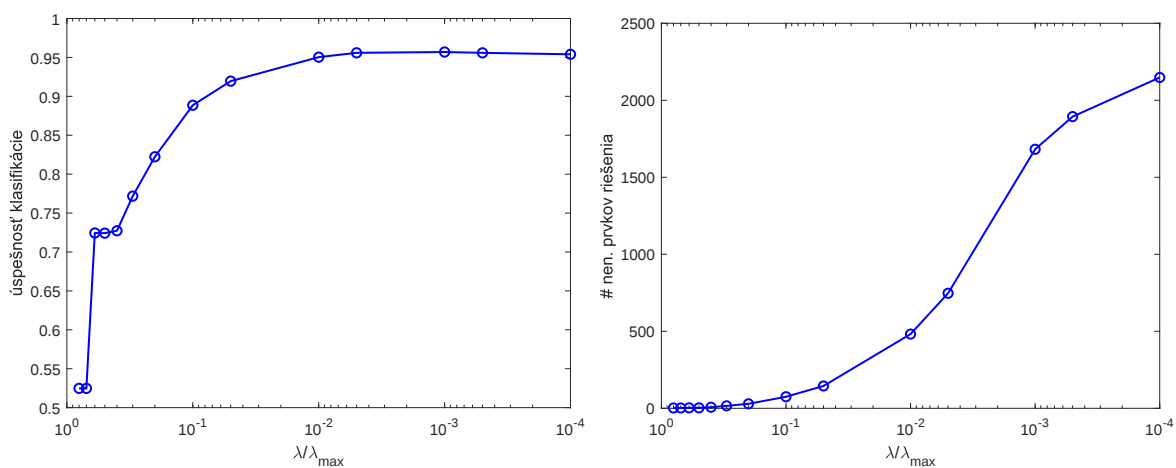
nenulových prvkov. Rozhodli sme sa otestovať aj hodnoty $0.01\lambda_{max}$, $0.005\lambda_{max}$, keďže boli výsledky medzi týmito dvomi hodnotami podobné, v práci uvádzame iba výsledok pre hodnotu $\lambda = 0.005\lambda_{max}$, pri ktorej má riešenie úspešnosť klasifikácie 95.59% a okolo 750 nenulových prvkov. V prípade dát Url sú rozumné hodnoty parametra λ z množiny $\{0.6\lambda_{max}, 0.1\lambda_{max}, 0.05\lambda_{max}\}$. Pre hodnotu $\lambda = 0.05\lambda_{max}$ sa dosahuje celkovo maximálna úspešnosť klasifikácie 96.15%, pričom riešenie má iba 15 nenulových prvkov. Testovali sme aj zvyšné dve hodnoty, hodnota $\lambda = 0.1\lambda_{max}$ poskytovala výsledky veľmi podobné ako pre $\lambda = 0.05\lambda_{max}$, preto ich v práci neuvádzame. Riešenie pre hodnotu parametra $\lambda = 0.6\lambda_{max}$ dosahuje na testovacej vzorke úspešnosť klasifikácie 94.67% a to iba s 3 (!) nenulovými prvkami, pričom rozmery dát sú 60000×3231949 . Pre dáta News sme zvolili hodnotu $\lambda = 0.01\lambda_{max}$, pri ktorej dosahujeme úspešnosť klasifikácie nad 93% (hodnota pochádza z 4-fold cross-validácie) a riešenie má okolo 720 nenulových prvkov. V prípade dát Webspam volíme hodnotu $\lambda = 0.01\lambda_{max}$, pri ktorej dosahujeme na testovacej vzorke úspešnosť klasifikácie nad 92.9% a riešenie má iba 61 nenulových prvkov. Pri týchto dvoch posledných dátových setoch sme volili hodnotu λ , ktorá dosahuje vysokú úspešnosť klasifikácie a riešenie pri nej neobsahuje veľa nenulových prvkov, aby sme mali menej signifikantných charakteristík. Poznamenajme, že pri dátach Webspam by počítanie pre menšie hodnoty λ zaberalo príliš veľké množstvo času navyše, čo sa z hľadiska pridanej úspešnosti klasifikácie neoplatí.



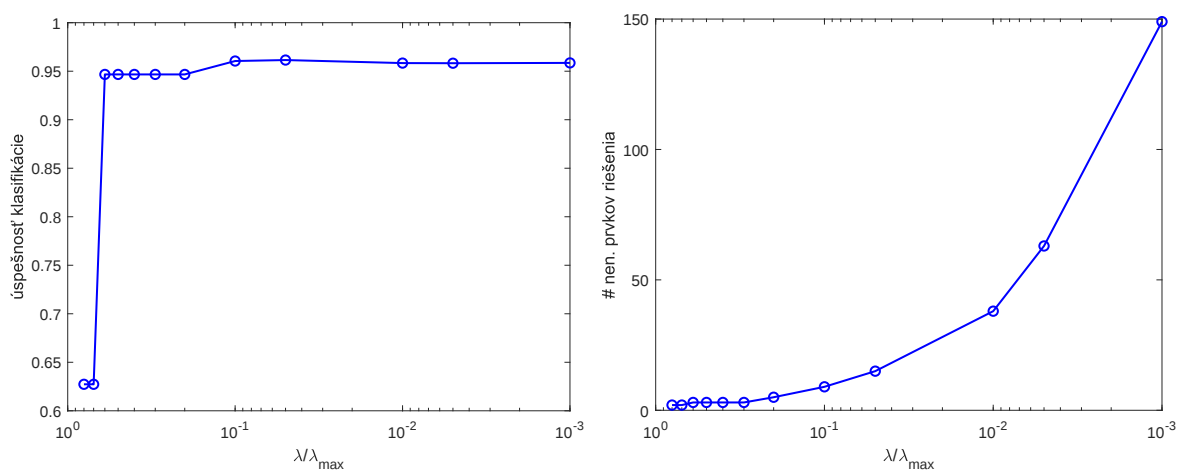
Obr. 7.11: Vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dáta Colon.



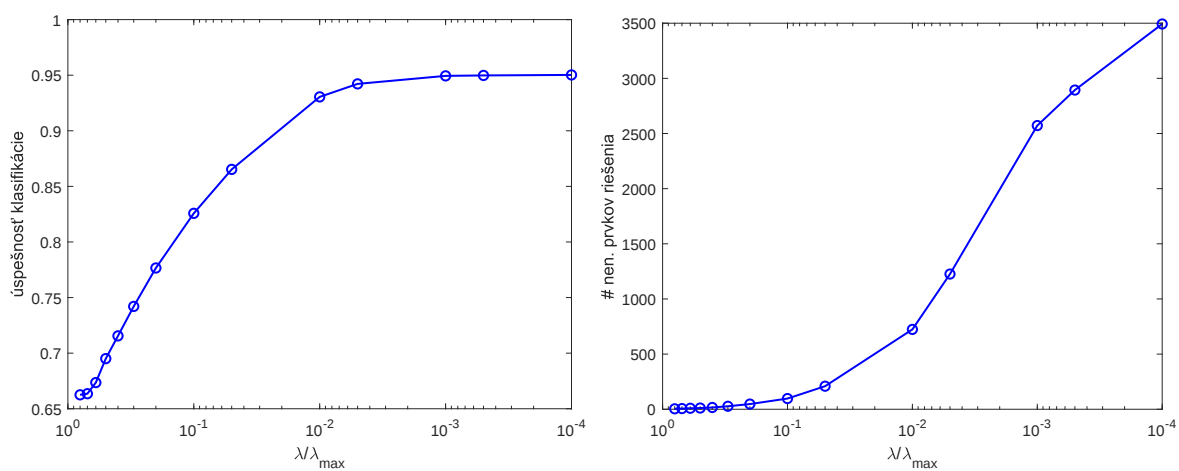
Obr. 7.12: Vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dáta Leukemia.



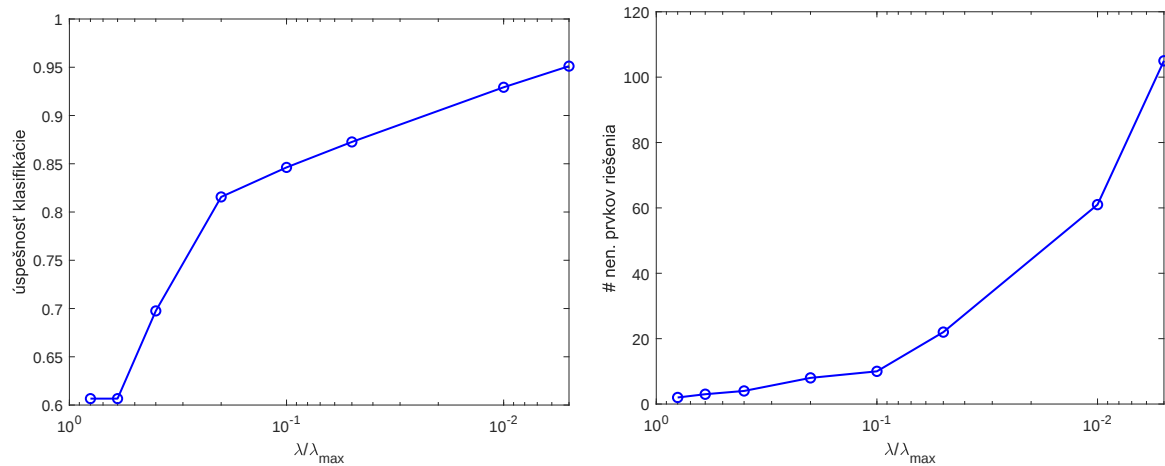
Obr. 7.13: Vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dáta Rcv1.



Obr. 7.14: Vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dáta Url.



Obr. 7.15: Vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dáta News.



Obr. 7.16: Vplyv hodnoty regularizačného parametra λ na úspešnosť klasifikácie a počet nenulových prvkov optimálneho riešenia pre dáta Webspam.

7.3.2 Porovnanie metód

V nasledujúcej časti porovnáme vybrané metódy pre riešenie úlohy l_1 -regulovanej logistickej regresie (1.7) na dátových setoch Colon, Leukemia, Rcv1, Url, News a Webspam pri hodnotách regularizačného parametra λ , ktoré sme určili v predošlej časti. Metódy tak testujeme za reprezentatívnych podmienok pre klasifikáciu pomocou l_1 -regulovanej logistickej regresie.

Pre tieto experimenty sme vybrali metódy podľa ich výsledkov z našich numerických experimentov na umelých dátach a takisto na základne predbežných výpočtov. Metódy PD11 a PD12, ktoré riešia Newtonov systém priamo, podávali omnoho horší výkon ako ich ekvivalenty PD11i a PD12i využívajúce iteračné metódy. Takisto v našich numerických experimentoch boli metódy APGs, DPG1 a PGs pravidelne porážené inými proximálnymi metódami. Preto spomínané metódy nezahrnieme do porovnania. Z proximálnych metód sme teda použili metódy APG, DPG1ls, DPG2, DPG2ls, DPG3, DPG3ls. Z primárno-duálnych metód vnútorného bodu sme použili PD11i, PD12i, PD13 a PD11ir s redukciou dimenzie. Výkon metód budeme porovnávať pri troch presnostiach 10^{-3} , 10^{-6} , 10^{-9} .

Pri PDMVB používame v nasledujúcich experimentoch bežné nastavenia (6.1), (6.2), (6.3) s dvomi výnimkami - používame parameter $\nu = 4$ pri všetkých úlohách

a parameter $\xi = 0.05$ používame pre menšie úlohy (Colon, Leukemia) a parameter $\xi = 0.001$ volíme pre ostatné úlohy. Parameter ξ riadi ako presne iteračnými metódami riešime Newtonove systémy.

Výsledky vybraných metód pri riešení úlohy l_1 -regulovanej logistickej regresie (1.7) pre dátové sety Colon, Leukemia, Rcv1, Url, News a Webspam možno pozorovať v Tabuľkách 7.13 - 7.20, v ktorých sledujeme čas a počet iterácii, ktoré metódy potrebovali na vypočítanie riešenia s relatívnou suboptimalitou 10^{-3} , 10^{-6} a 10^{-9} . V týchto tabuľkách takisto uvádzame počet nenulových zložiek týchto dosiahnutých riešení. V prípade PDMVB metód bolo použité zaokrúhľovacie pravidlo (7.2) (keďže bežné riešenia pomocou PDMVB majú všetky zložky nenulové). Zaujímavou otázkou je aj, aký vplyv má presnosť daných riešení na úspešnosť následnej klasifikácie. Preto v tabuľkách pre dáta, pri ktorých máme testovaciu vzorku, uvádzame aj úspešnosť klasifikácie pomocou riešenia pri danej presnosti.

V Tabuľkách 7.13 - 7.20 môžeme vidieť, že proximálne metódy s určenou dĺžkou kroku v každej iterácii podávajú značne lepšie výsledky ako ich verzie pre konštantnú dĺžku kroku. Taktiež možno vidieť, že metóda PD11ir šetrí oproti pôvodnej metóde PD11i veľa výpočtového času a iterácii. Na dvoch najťažších riešiteľných problémoch dosahuje viac ako 8-násobné zrýchlenie výpočtu.

Z Tabuliek 7.13 a 7.14 možno pozorovať, že väčšina metód potrebovala na vyriešenie daných problémov značne menej ako 1 sekundu, jedná sa o pomerne jednoducho riešiteľné problémy. Najlepšie obstáli metódy DPG2ls, DPG3ls, PD11ir a PD13. Pri ostatných veľkorozmernejších úlohách však už metóda PD13, ktorá rieši Newtonov systém priamo, nepodávala dobrý výkon, preto ju v ostatných tabuľkách neuvádzame.

V Tabuľkách 7.15-7.18 si väčšinou najlepšie viedli metódy PD11ir, DPG2ls, DPG3ls, PD11i. Vo všetkých prípadoch okrem Tabuľky 7.18 bola pri výpočte riešenia s relatívnou suboptimalitou 10^{-9} najrýchlejšia PD11ir.

V Tabuľke 7.19 jednoznačne najlepší výkon podáva metóda PD11ir, ktorá problém na dátach News vyriešila za 35 sekúnd, čo predstavuje lepší výpočtový čas ako časy, ktoré ostatné metódy potrebovali na dosiahnutie prvej presnosti. Na tejto úlohe pomocou metódy PD11ir dosahujeme 8-násobne rýchlejšie nájdenie riešenia s relatívnou suboptimalitou 10^{-9} ako v prípade metódy PD11i, z ktorej vznikla.

Tabuľka 7.20 prezentuje porovnanie najlepších metód (APG, DPG1ls, DPG2ls, DPG3ls, PD11i a PD11ir) na dátach Webspam pre hodnotu regularizačného parametra $\lambda = 0.01\lambda_{max}$. V prípade tejto úlohy sa jedná o celkovo najťažšie riešiteľný problém v tejto práci - samotné dáta, na ktorých prebiehala optimalizácia, zaberali viac ako 11GB v operačnej pamäti počítača. Každý metóde sme dali na vyriešenie úlohy 3 hodiny. Za tento čas túto úlohu pre finálnu presnosť 10^{-9} boli schopné vyriešiť len 3 metódy. Metóda DPG2ls na to potrebovala viac ako 178 minút, metóda PD11i potrebovala 110 minút. Jenoznačne najlepšie obstála metóda PD11ir, ktorá úlohu vyriešila len za 13 minút. S týmto výpočtovým časom prekonala aj časy, ktoré väčšina metód potrebovala na dosiahnutie prvej presnosti. Oproti pôvodnej metóde PD11i sme tak riešenie dosiahli viac ako 8-násobne rýchlejšie a metóda PD11ir pri tom potrebovala o 20 iterácii menej. V poslednom stĺpci Tabuľky 7.20 uvádzame pri metódach, ktoré za 3 hodiny nestihli úlohu dopočítať, relatívnu suboptimalitu ich finálneho riešenia.

V Tabuľkách 7.13 - 7.20 si možno všimnúť, že počty nenulových zložiek riešení s relatívnou suboptimalitou 10^{-9} sú medzi metódami viacmenej rovnaké, možno pozorovať len malé rozdiely. Poznamenajme, že hodnota počtu nenulových zložiek riešenia je často o niečo menšia u metódy PD11ir, avšak tento jav je spôsobený tým, že sme na všetky PDMVB metódy aplikovali zaokrúhľovacie pravidlo (7.2). Metóda PD11ir toto pravidlo úplne nepotrebuje - hlavne pre väčšie presnosti, pretože vtedy už podáva riešenie s výrazne riedkou štruktúrou narozdiel od klasických PDMVB, kde sú všetky zložky riešenia nenulové. Keďže sme túto metódu však v tabuľkách nechceli znevýhodňovať oproti iným PDMVB, použili sme na určenie počtu nenulových zložiek to isté pravidlo. Výsledné počty nenulových zložiek riešenia pri metóde PD11ir, ak by sme zokrúhľovacie pravidlo (7.2) nepoužili možno sledovať v Tabuľke 7.21. Táto tabuľka nám hovorí aj o tom, ako metóda postupne redukovala dimenziu dát, s ktorými pracovala. Môžeme si všimnúť, že už pri prvej presnosti je dimenzia dát častokrát výrazne redukovaná, čo vysvetľuje veľkú úsporu výpočtového času oproti pôvodnej metóde PD11i. Čím má metóda PD11ir presnejšie riešenie, tým má viac redukovanú dimenziu dát a optimalizáciu sústreďuje už iba na podstatné nenulové zložky riešenia. Týmto spôsobom vieme dostať presnejšie riešenia (aj pre $\epsilon < 10^{-9}$) za pomerne malý čas navyše, čo je veľkou výhodou metódy PD11ir.

Pri našom experimente sledovania vplyvu presností riešení (používame pôvodné nezaokrúhľované riešenia) na úspešnosť následnej klasifikácie, sme prišli na zaujímavé výsledky. Riešenia s relatívnou suboptimalitou 10^{-3} pochádzajúce z rôznych metód dosahujú rôzne výsledné úspešnosti klasifikácie. V prípade Tabuľky 7.14 pre dáta Leukemia a Tabuľky 7.17 pre Url ($\lambda = 0.6\lambda_{max}$) je tento jav najextrémnejší. V Tabuľke 7.14 možno pozorovať, že riešenia pomocou PDMVB majú úspešnosť klasifikácie 85.29%, zatiaľ čo riešenie pomocou DPG1ls dosahuje iba úspešnosť klasifikácie 79.41% a ostatné proximálne metódy dosahujú úspešnosť klasifikácie 82.35%. V Tabuľke 7.17 riešenia pomocou proximálnych metód APG, DPG2, DPG3 dosahujú úspešnosť klasifikácie len 62.73% (!), pritom však riešenia pomocou ostatných metód tej istej presnosti majú úspešnosť klasifikácie 94.67%. Výsledné úspešnosti klasifikácie pre riešenia s relatívnou suboptimalitou 10^{-6} sú už pri všetkých metódach rovnaké na všetkých dátových setoch. Pri riešeníach s relatívnou suboptimalitou 10^{-9} sa tieto hodnoty úspešností už takisto nemenili, preto tento stĺpec v tabuľkách neuvádzame. Presnejšie riešenia ako riešenia s relatívnou suboptimalitou 10^{-6} mali pri našich dátach význam predovšetkým na presnejšie určenie pravých nenulových zložiek riešenia (t.j. presnejší výber signifikantných charakteristík) pri danej hodnote parametra λ . Z týchto predošlých výsledkov experimentov však možno usúdiť, že má význam hľadať presnejšie riešenia.

Na Obr. 7.17 možno pozorovať porovnanie vývoja relatívnej suboptimality v čase pre metódy DPG2ls, DPG3ls, PD11i a PD11ir na dátovom sete Rcv1 pri voľbe regularizačného parametra $\lambda = 0.005\lambda_{max}$ a $\lambda = 0.001\lambda_{max}$. Vidíme, že voľba parametra λ má veľmi výrazný vplyv na výkon všetkých metód, úloha s menšou hodnotou $\lambda = 0.001\lambda_{max}$ je ťažšie riešiteľná. Pri metóde PD11ir na obrázkoch vidieť, že metóda má spočiatku podobný výkon ako PD11i, no akonáhle začne redukovať dimenziu, začne výrazne šetriť výpočtový čas. Pri výpočte riešení s relatívnou suboptimalitou menšou ako 10^{-7} obstála metóda PD11ir najlepšie. Možno si všimnúť, že za výsledný čas, ktorý uvádzame v tabuľkách, metóda PD11ir v skutočnosti poskytla ešte presnejšie riešenie ako ostatné metódy. Takisto možno pozorovať, že pre $\lambda = 0.001\lambda_{max}$ sú metódy DPG2ls, DPG3ls a PD11ir veľmi vyrovnané až po relatívnu suboptimalitu riešení okolo 10^{-7} .

Na Obr. 7.18 porovnáваме vývoj relatívnej suboptimality v čase pre metódy DPG2ls, DPG3ls, PD11i a PD11ir na dátových setoch News a Webspam. Jedná sa o dva najťažšie riešiteľné problémy v tejto časti. Pri metóde PD11ir opäť možno pozorovať, že spočiatku podáva podobný výkon ako PD11i, avšak v ďalších fázach podáva táto metóda omnoho lepší výkon vďaka postupnej redukcii dimenzie dát a sústredeniu optimalizácie na podstatné nenulové zložky riešenia. Najlepší výkon jednoznačne podáva metóda PD11ir s redukciou dimenzie, najlepší výpočtový čas má už pri relatívnej suboptimality 10^{-3} na oboch dátových setoch. Takisto možno pozorovať, že metódy DPG2ls, DPG3ls a PD11i podávajú veľmi zrovnateľný výkon pri dátach News do relatívnej suboptimality 10^{-4} a pri dátach Webspam do relatívnej suboptimality 10^{-5} .

Tabuľka 7.13: Porovnanie metód na dátach Colon pre $\lambda = 0.1\lambda_{max}$.

Presnosť ϵ	10^{-3}			10^{-6}			10^{-9}		
	čas	it.	nnz	čas	it.	nnz	čas	it.	nnz
APG	0,0076	293	30	0,0512	2108	26	0,2449	10483	26
DPG1ls	0,0025	52	28	0,0072	137	26	0,0256	448	26
DPG2	0,0145	355	28	0,0419	1008	26	0,0693	1671	26
DPG2ls	0,0027	52	28	0,0044	83	26	0,0062	120	26
DPG3	0,0185	491	27	0,0513	1256	26	0,0847	2032	26
DPG3ls	0,0030	52	28	0,0049	85	26	0,0069	125	26
PD11i	0,0073	14	17	0,0117	19	26	0,0158	24	26
PD11ir	0,0069	13	23	0,0073	16	24	0,0074	18	26
PD12i	0,0301	14	13	0,0487	20	26	0,0624	25	26
PD13	0,0076	15	19	0,0100	20	26	0,0124	25	26

Tabuľka 7.14: Porovnanie metód na dátach Leukemia pre $\lambda = 0.01\lambda_{max}$.

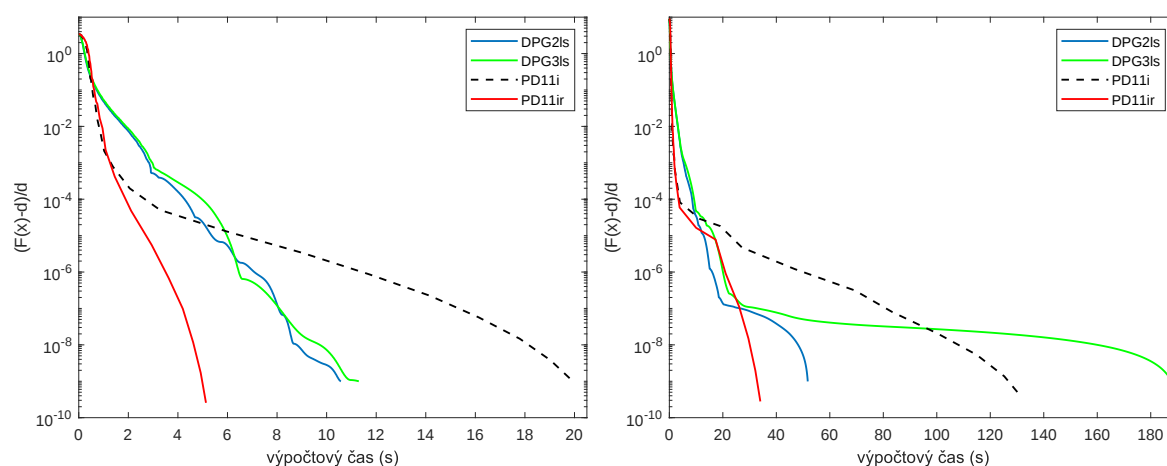
Presnosť ϵ												
		10^{-3}				10^{-6}				10^{-9}		
Metóda	čas	it.	nnz	úsp.	čas	it.	nnz	úsp.	čas	it.	nnz	
APG	0,300	4341	31	82,35	1,911	27949	18	85,29	13,602	198263	18	
DPG1ls	0,014	115	23	79,41	0,041	322	18	85,29	0,117	890	18	
DPG2	0,771	7547	22	82,35	1,554	15179	18	85,29	2,227	21652	18	
DPG2ls	0,015	114	23	82,35	0,023	173	18	85,29	0,029	217	18	
DPG3	1,167	11376	21	82,35	2,319	22629	18	85,29	3,112	30254	18	
DPG3ls	0,015	117	23	82,35	0,024	184	18	85,29	0,032	246	18	
PD11i	0,068	26	12	85,29	0,087	31	18	85,29	0,103	37	18	
PD11ir	0,060	25	18	85,29	0,061	27	18	85,29	0,062	31	18	
PD12i	0,285	28	14	85,29	0,336	33	18	85,29	0,371	38	18	
PD13	0,042	24	17	85,29	0,052	29	18	85,29	0,067	35	18	

Tabuľka 7.15: Porovnanie metód na dátach Rcv1 pre $\lambda = 0.005\lambda_{max}$.

Tabuľka 1: Presnosť algoritmu APG na dátach PD12 pre $\epsilon = 10^{-3}, 10^{-6}, 10^{-9}$											
Presnosť ϵ	10^{-3}				10^{-6}				10^{-9}		
Metóda	čas	it.	nnz	úsp.	čas	it.	nnz	úsp.	čas	it.	nnz
APG	3,75	1596	828	95,58	23,87	10122	748	95,59	148,20	62870	747
DPG1ls	2,81	565	799	95,58	23,53	4717	750	95,59	128,46	25831	747
DPG2	7,25	1596	828	95,58	18,99	4168	746	95,59	28,86	6328	747
DPG2ls	2,84	565	799	95,58	7,08	1411	746	95,59	10,58	2112	747
DPG3	7,23	1596	828	95,58	23,00	4943	746	95,59	33,04	7078	747
DPG3ls	2,97	565	799	95,58	6,46	1237	747	95,59	11,30	2167	747
PD11i	1,38	35	266	95,60	11,72	40	745	95,59	19,95	45	747
PD11ir	1,45	34	285	95,60	3,63	37	727	95,59	5,13	41	746
PD12i	4,87	37	328	95,59	25,26	42	745	95,59	35,22	47	747

Tabuľka 7.16: Porovnanie metód na dátach Rcv1 pre $\lambda = 0.001\lambda_{max}$.

Presnosť ϵ											
Metóda	10^{-3}				10^{-6}				10^{-9}		
	čas	it.	nnz	úsp.	čas	it.	nnz	úsp.	čas	it.	nnz
APG	11,3	4790	2177	95,69	93,9	39184	1685	95,69	755,1	314661	1682
DPG1ls	5,3	1039	1884	95,69	42,9	8691	1686	95,69	284,3	57650	1682
DPG2	24,3	5339	1806	95,69	73,0	16092	1686	95,69	221,2	49088	1681
DPG2ls	5,3	1065	1814	95,68	16,0	3204	1686	95,69	51,7	10362	1683
DPG3	24,7	5353	1792	95,69	97,3	21058	1686	95,69	305,6	67787	1682
DPG3ls	5,9	1161	1766	95,68	20,0	4018	1685	95,69	187,8	38080	1682
PD11i	2,0	45	728	95,70	69,4	52	1675	95,69	131,9	57	1682
PD11ir	1,9	44	719	95,70	21,0	49	1643	95,69	34,0	53	1680
PD12i	5,5	48	745	95,70	91,0	55	1671	95,69	266,1	62	1682

Obr. 7.17: Porovnanie vývoja relatívnej suboptimality v čase pre metódy DPG2ls, DPG3ls, PD11i, PD11ir na dátach Rcv1 pre $\lambda = 0.005\lambda_{max}$ a $\lambda = 0.001\lambda_{max}$.

Tabuľka 7.17: Porovnanie metód na dátach Url pre $\lambda = 0.6\lambda_{max}$.

Presnosť ϵ	10^{-3}				10^{-6}				10^{-9}		
	čas	it.	nnz	úsp.	čas	it.	nnz	úsp.	čas	it.	nnz
APG	1,72	53	3	62,73	25,63	790	3	94,67	92,57	2853	3
DPG1ls	1,81	28	3	94,67	5,45	95	3	94,67	25,00	457	3
DPG2	2,50	53	3	62,73	37,16	792	3	94,67	40,01	853	3
DPG2ls	1,81	28	3	94,67	5,64	93	3	94,67	8,02	135	3
DPG3	2,47	53	3	62,73	40,15	858	3	94,67	43,13	922	3
DPG3ls	1,82	28	3	94,67	6,75	117	3	94,67	10,36	178	3
PD11i	5,30	9	2	94,67	9,16	15	3	94,67	12,11	20	3
PD11ir	5,45	8	1	94,67	5,53	10	2	94,67	5,54	12	3
PD12i	11,92	8	2	94,67	18,01	13	3	94,67	24,96	19	3

Tabuľka 7.18: Porovnanie metód na dátach Url pre $\lambda = 0.05\lambda_{max}$.

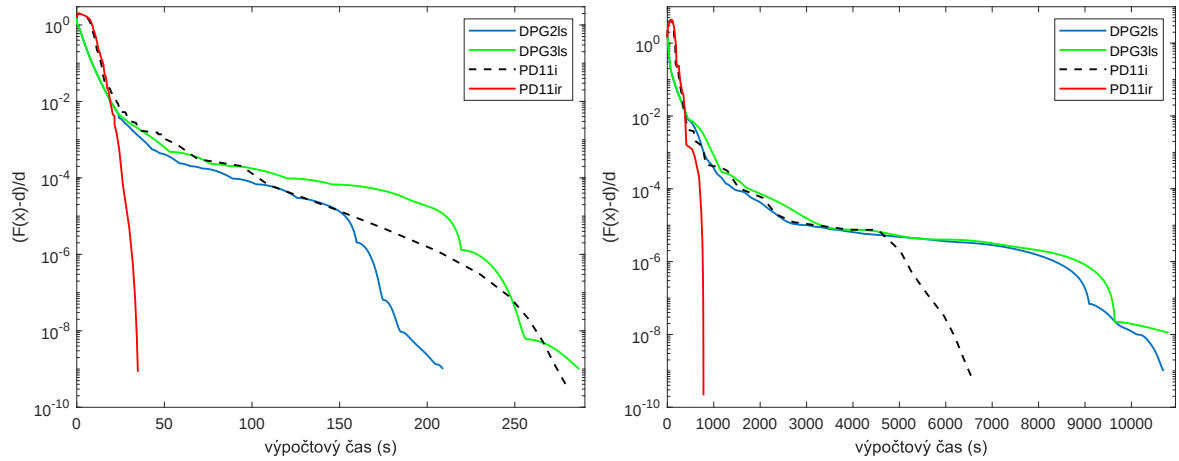
Presnosť ϵ	10^{-3}				10^{-6}				10^{-9}		
	čas	it.	nnz	úsp.	čas	it.	nnz	úsp.	čas	it.	nnz
APG	7,82	241	24	96,16	45,19	1393	15	96,15	293,28	9052	15
DPG1ls	3,89	66	20	96,13	11,68	208	15	96,15	26,17	474	15
DPG2	14,12	299	24	96,15	53,20	1131	15	96,15	80,44	1710	15
DPG2ls	3,90	66	20	96,13	9,56	163	15	96,15	11,80	201	15
DPG3	15,73	336	24	96,15	56,44	1207	15	96,15	73,45	1570	15
DPG3ls	3,88	66	20	96,13	9,16	161	15	96,15	11,62	205	15
PD11i	13,00	18	8	96,15	22,68	26	15	96,15	29,01	32	15
PD11ir	13,45	17	11	96,13	13,56	19	10	96,15	13,61	22	15
PD12i	43,06	11	10	96,14	68,79	17	15	96,15	85,32	22	15

Tabuľka 7.19: Porovnanie metód na dátach News pre $\lambda = 0.01\lambda_{max}$.

Presnosť ϵ	10^{-3}			10^{-6}			10^{-9}		
	čas	it.	nnz	čas	it.	nnz	čas	it.	nnz
APG	43,7	1841	943	330,7	13885	725	2584,9	108772	724
DPG1ls	38,2	894	857	299,0	7020	724	1938,5	45620	724
DPG2	69,0	1824	895	351,9	9303	723	436,7	11543	724
DPG2ls	37,5	868	813	167,1	3901	723	209,0	4879	724
DPG3	73,8	1954	808	523,6	13861	723	686,2	18169	724
DPG3ls	44,5	1039	789	228,5	5360	723	286,7	6729	724
PD11i	57,2	83	325	229,9	90	724	279,7	95	725
PD11ir	24,2	58	415	32,0	61	675	34,9	64	723
PD12i	135,8	77	244	541,8	87	724	661,2	92	725

Tabuľka 7.20: Porovnanie metód na dátach Webspam pre $\lambda = 0.01\lambda_{max}$.

Presn. ϵ	10^{-3}				10^{-6}				10^{-9}			$3h$
	čas	it.	nnz	úsp.	čas	it.	nnz	úsp.	čas	it.	nnz	
APG	809	612	94	92,92	9752	7364	62	92,92	—	8157	61	7,2E-07
DPG1ls	690	281	82	92,94	7877	3236	62	92,92	—	4438	61	1,2E-07
DPG2ls	786	316	79	92,89	8356	3429	62	92,92	10695	4388	61	
DPG3ls	969	395	80	92,91	8866	3642	62	92,92	—	4437	61	1,1E-08
PD11i	828	41	30	92,95	5272	51	60	92,92	6581	56	61	
PD11ir	593	29	29	92,97	767	33	53	92,92	778	36	61	

Obr. 7.18: Porovnanie vývoja relatívnej suboptimality v čase pre metódy DPG2ls, DPG3ls, PD11i, PD11ir na dátach News a dátach Webspam pre $\lambda = 0.01\lambda_{max}$.

Presnosť ϵ		10^{-3}	10^{-6}	10^{-9}
Dáta	λ	nnz	nnz	nnz
Colon	$0.1\lambda_{max}$	54	28	26
Leukemia	$0.01\lambda_{max}$	216	37	18
Rcv1	$0.005\lambda_{max}$	6678	1027	767
Rcv1	$0.001\lambda_{max}$	24519	3042	1758
Url	$0.6\lambda_{max}$	3	3	3
Url	$0.05\lambda_{max}$	22	18	15
News	$0.01\lambda_{max}$	18048	1135	753
Webspam	$0.01\lambda_{max}$	776	100	62

Tabuľka 7.21: Výsledné počty nenulových zložiek riešenia pri metóde PD11ir bez zaokrúhlovacieho pravidla (7.2).

Záver

Dizertačná práca sa zaoberala modernými metódami konvexnej optimalizácie vhodnými na riešenie nediferencovateľných úloh na voľný extrém špeciálnej štruktúry, pri ktorých je účelová funkcia rozložiteľná na konvexnú diferencovateľnú a nediferencovateľnú časť. Cieľom práce bolo spracovať prehľad známych metód riešenia, ich teoretická analýza, navrhnutie prípadných modifikácií či numerických vylepšení metód. Medzi ciele práce takisto patrilo aj testovanie a porovnanie jednotlivých metód riešenia vykonaním numerických experimentov na vygenerovaných dátach a aplikácia vybraných metód na reálne problémy z rôznych oblastí.

V Kapitole 1 sme sa venovali úlohám na voľný extrém so špeciálnou štruktúrou, pričom sme uviedli viaceré aplikácie týchto úloh a metódy riešenia. Konkrétne sme sa zaoberali tzv. l_1 -regulovanými úlohami, ktoré obsahujú vo svojej účelovej funkcii l_1 normu a majú v praxi široké uplatnenie. Medzi najvýznamnejších predstaviteľov týchto úloh patrí úloha l_1 -regulovaných najmenších štvorcov a úloha l_1 -regulovanej logistickej regresie, na ktoré sme sa sústredili v ďalšej časti práce. Pre tieto úlohy sme spravili prehľad článkov iných autorov o danej problematike, kde sme spracovali prehľad aplikácii a takisto prehľad používaných metód riešenia.

Ďalšie dve kapitoly práce boli venované moderným metódam konvexnej optimalizácie použiteľným na riešenie l_1 -regulovaných úloh. Konkrétne sme sa zaoberali proximálnymi metódami a metódami vnútorného bodu. Kapitola 2 bola venovaná proximálnym metódam, predstavili sme v nej postupne proximal-gradientnú metódu, zrýchlenú proximal-gradientnú metódu, proximal-newtonovu metódu, prírastkovú proximal-gradientnú metódu a nakoniec ADMM metódu. Taktiež sme navrhli novú proximal-kvázinewtonovu metódu a na základe rôznych interpretácií proximal-gradientnej metódy sme odvodili analogické interpretácie pre proximal-newtonovu metódu a proximal-

kvázinewtonovu metódu. Pomocou týchto rôznych interpretácií sme na mnohé proximálne metódy poskytli zjednocujúci pohľad. V Kapitole 2 sme takisto predstavili 3 spádové modifikácie zrýchlenej proximal-gradientnej metódy, z toho jedna je našim výtvorom. Jedná sa o hybridnú metódu, ktorá vo svojom algoritme spája zrýchlenú proximal-gradientnú metódu a základnú proximal-gradientnú metódu, ktorá je spádová.

Kapitola 3 sa venovala metódam vnútorného bodu, kde sme uviedli primárno-duálnu metódu vnútorného bodu, jej myšlienku, primárno-duálny smer optimalizácie a algoritmus. V tejto kapitole sme takisto predstavili novú primárno-duálnu metódu vnútorného bodu s redukciou dimenzie, ktorú sme vytvorili špeciálne na riešenie l_1 -regulovaných úloh. Táto hybridná metóda kombinuje metódy vnútorného bodu s proximálnymi metódami a prináša metódam vnútorného bodu niekoľko potenciálnych výhod, medzi ktoré patrí predovšetkým zrýchlenie výpočtového času. Vysvetlili sme myšlienku metódy, objasnili jej fungovanie a takisto sme uviedli schému algoritmu.

V Kapitole 4 sme sa zaoberali implementáciou proximálnych metód a primárno-duálnych metód vnútorného bodu priamo na riešenie úloh l_1 -regulovaných najmenších štvorcov a l_1 -regulovanej logistickej regresie. V časti o implementácii proximálnych metód sme odvodili tvary proximálnych operátorov pre naše úlohy a takisto sme sa venovali problematike voľby dĺžky kroku. Keďže metódy vnútorného bodu sú používané na riešenie diferencovateľných konvexných optimalizačných úloh, v časti o implementácii metód vnútorného bodu sme uviedli 2 spôsoby transformácie l_1 -regulovaných úloh na konvexnú úlohu s diferencovateľnou účelovou funkciou a lineárnymi ohraničeniami. Pre obe tieto ekvivalentné úlohy sme odvodili Newtonove systémy pre hľadanie primárno-duálnych smerov optimalizácie. Takisto sme spomenuli a odvodili možné vylepšenia využiteľné pri riešení daných systémov.

Rozsiahlym numerickým experimentom na umelo generovaných dátach sa venovala Kapitola 6, kde sme testovali výkon veľkého množstva nami naprogramovaných metód v prostredí MATLAB, pričom sme sa snažili čo najobjektívnejšie zhodnotiť výkon metód, najmä z hľadiska výpočtového času. Tieto numerické experimenty považujeme za jeden z hlavných prínosov práce. Metódy sme porovnávali na generovaných úlohách l_1 -regulovaných najmenších štvorcov ako aj na generovaných úlohách l_1 -regulovanej

logistickej regresie a to pre viaceré spôsoby generovania dát, čo podľa nášho názoru ešte zvýšilo objektivnosť nášho porovnania metód. Na umelo generovaných úlohách l_1 -regulovaných najmenších štvorcov sme porovnávali výkon metód pri zmene rôznych parametrov generovaných úloh. Skúmali sme vplyv regularizačného parametra λ , vplyv oboch rozmerov m a n , vplyv riedkosti matice dát, vplyv vzdialenosti počítateľného riešenia od optima, vplyv počtu nenulových prvkov optima, vplyv dynamického rozsahu optima a vykonali sme test škálovateľnosti metód, podobným spôsobom ako v článku [56]. Takisto na umelo generovaných úlohách l_1 -regulovanej logistickej regresie sme porovnávali výkon metód pri zmene rôznych parametrov generovaných úloh. Skúmali sme vplyv regularizačného parametra λ , vplyv oboch rozmerov m a n , vplyv riedkosti matice dát a vykonali sme test škálovateľnosti metód, podobným spôsobom ako v článku [58]. Medzi testovanými metódami bola aj naša spádová modifikácia zrýchlenej proximal-gradientnej metódy s modifikovaným algoritmom určovania dĺžky kroku DPG2ls a takisto naša nová primárno-duálna metóda vnútorného bodu s redukciou dimenzie PD11ir. Tieto metódy podávali na umelo generovaných dátach výborné výsledky. Za celkovo „univerzálne najlepšiu“ metódu považujeme práve metódu PD11ir, ktorá dosahovala pre väčšinu experimentov na oboch generovaných úlohách najlepšie výpočtové časy pre najvyššie požadované presnosti pri ťažšie riešiteľných úlohách z daných experimentov. Oproti pôvodnej PDMVB metóde PD11i, z ktorej PD11ir vznikla, sme pri každom experimente pozorovali zlepšenie vo forme úspory počtu iterácií, ale predovšetkým v úspore výpočtového času. Demonštrovali sme tak prínos pridania redukcie dimenzie dát pomocou proximálnych metód do algoritmu primárno-duálnej metódy vnútorného bodu. V experimentoch sme taktiež poukázali na mnohé vlastnosti algoritmov metód.

V Kapitole 7 sme sa venovali reálnym aplikáciám l_1 -regulovaných úloh, testovali sme funkčnosť vybraných metód riešenia na reálnych problémoch (a dátach) z rôznych oblastí. Metódy sme porovnávali na problémoch: dizajn mechanicej konštrukcie (TTD), rekonštrukcia riedkeho signálu (SSR) a l_1 -regulovaná logistická regresia. V prípade TTD úloh z oblasti optimálneho topologického dizajnu najlepšie výpočtové časy dosahovali PDMVB a predovšetkým hybridná metóda HYB2, ktorú sme zostrojili na základe výsledkov iných metód. Metóda HYB2 vo svojom algoritme kombinuje tri me-

tódy a využíva myšlienku redukcie dimenzie. Pri porovnaní metód na SSR úlohách z oblasti spracovania signálov bolo zrejmé, že dané dáta vyhovujú proximálnym metódam. Najlepšie výpočtové časy dosiahli spádové modifikácie zrýchlenej proximal-gradientnej metódy s modifikovaným algoritmom určovania dĺžky kroku DPG2ls a DPG3ls. Pri testovaní funkčnosti metód na riešenie l_1 -regulovanej logistickej regresie sme používali reálne dáta pochádzajúce z medicíny, z oblasti kategorizácie textu a z oblasti detekovania škodlivých stránok a spamu. Keďže sme chceli metódy riešenia úloh otestovať za reprezentatívnych podmienok a relevantnú hodnotu regularizačného parametra λ pre klasifikáciu pomocou l_1 -regulovanej logistickej regresie, rozhodli sme sa vykonať na dátach analýzu vplyvu parametra λ . Pričom sme sa tejto analýze venovali pre prípad, keď máme testovaciu vzorku aj keď testovaciu vzorku nemáme. Pre dáta bez testovacej vzorky sme na určenie hodnoty λ používali tzv. k-fold cross-validáciu, pričom sme využili PDMVB. Pri následnom porovnaní výkonu metód dosiahla celkovo najlepšie výsledky PDMVB metóda PD11ir. V testoch na jednotlivých úlohách bola konzistentne medzi najlepšími metódami, resp. vo väčšine úloh podala najrýchlejší výpočtový čas. Opäť sme tak mohli pozorovať prínos redukovania dimenzie pomocou poznatkov z proximálnych metód. V daných experimentoch sme taktiež skúmali vplyv presnosti pri našom stopovacom kritériu na počet nenulových prvkov dosiahnutého riešenia či na následnú úspešnosť klasifikácie pomocou dosiahnutého riešenia. Zistili sme, že má zmysel počítat riešenia do väčšej presnosti.

Boli sme veľmi spokojní s fungovaním našich metód PD11ir a DPG2ls v praxi a to na umelo generovaných úlohách ako aj pri reálnych aplikáciach. V prípade oboch metód sa jedná o hybridné metódy kombinujúce myšlienky a algoritmy rôznych metód. Takisto sme vytvorili veľmi efektívnu hybridnú metódu z 3 metód na riešenie TTD úloh. Preto si myslíme, že sme v práci ukázali, že má zmysel skúmať takéto hybridné metódy kombinujúce dobré vlastnosti rôznych algoritmov. Túto oblasť považujeme za oblasť ďalšieho potenciálneho výskumu. V dnešnej dobe vyžadujú mnohé aplikácie riešiť veľkorozmerné úlohy, preto si myslíme, že jeden z najpodstatnejších prínosov práce je práve vytvorenie metódy PD11ir, ktorá v prípade l_1 -regulovaných úloh vie túto dimenziu dať rozumne redukovať a sústrediť tak optimalizáciu na podstatné zložky riešenia. Jej funkčnosť sme experimentálne potvrdili na úlohách l_1 -regulovaných

najmenších štvorcov ako aj úlohách l_1 -regulovanej regresie a to pre umelé aj reálne dáta. Pri l_1 -regulovanej regresii na reálnych dátach z rôznych oblastí sme pomocou PD11ir vedeli riešiť veľkorozmerné úlohy do veľkej presnosti za vynikajúce výpočtové časy v porovnaní s inými metódami. Pomocou PD11ir sme úspešne vyriešili veľkorozmerné úlohy rozmerov 60000×3231949 , 19996×1355191 či 200000×680715 , pričom posledná menovaná zaberala v operačnej pamäti počítača viac ako 11GB. Otázku, či možno nejakým rozumným spôsobom redukovať dimenziu dát pri úlohách iného typu a urýchliť tak následnú optimalizáciu, považujeme tiež za oblasť ďalšieho potenciálneho výskumu.

Literatúra

- [1] Alon U., Barkai N., Notterman D. A., Gish K., Ybarra S., Mack D. and Levine A. J., *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, Proceedings of the National Academy of Sciences, 96(12), 6745-6750, (1999).
- [2] Atchade Y. F., Fort G. and Moulines E., *On stochastic proximal gradient algorithms*, arXiv preprint arXiv:1402.2365, (2014).
- [3] Barzilai J. and Borwein J. M., *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 8(1), 141-148, (1988).
- [4] Beck A. and Teboulle M., *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2(1), 183-202, (2009).
- [5] Beck A. and Teboulle M., *Gradient-based algorithms with applications to signal recovery*, Convex optimization in signal processing and communications, 42-88, (2009).
- [6] Becker S., Bobin J. and Candes E. J., *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM Journal on Imaging Sciences, 4(1), 1-39, (2011).
- [7] Ben-Tal A. and Nemirovski A., *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications.*, SIAM, Philadelphia, 488 p., (2001).
- [8] Ben-Tal A. and Nemirovski A., *Robust truss topology design via semidefinite programming*, SIAM Journal on Optimization, 7(4), 991-1016, (1997).

- [9] Bertsekas D. P., *Nonlinear programming*, Athena Scientific Optimization and Computation Series, Belmont, 791 p., (1999).
- [10] Bertsekas D. P., *Incremental gradient, subgradient, and proximal methods for convex optimization: A survey*, Optimization for Machine Learning 2010, 1-38, (2011).
- [11] Bertsekas D. P., *Incremental proximal methods for large scale convex optimization*, Mathematical programming, 129(2), 163-195, (2011).
- [12] Bishop C. M., *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, 738 p., (2006).
- [13] Blatt D., Hero A. O. and Gauchman H., *A convergent incremental gradient method with a constant step size*, SIAM Journal on Optimization, 18(1), 29-51, (2007).
- [14] Boyd S. and Vandenberghe L., *Convex optimization*, Cambridge University Press, Cambridge, 716 p., (2004).
- [15] Boyd S. and Vandenberghe L., *Subgradients*, Notes for EE364b, Stanford University, April, (2008).
- [16] Boyd S., Parikh N., Chu E., Peleato B. and Eckstein J., *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends in Machine Learning, 3(1), 1-122, (2011).
- [17] Broughton R., Coope I., Renaud P. and Tappenden R., *A Barzilai-Borwein l_1 -Regularized Least Squares Algorithm for Compressed Sensing*, arXiv preprint arXiv:0907.3340, (2009).
- [18] Bubeck S., *Convex Optimization: Algorithms and Complexity*, Foundations and Trends in Machine Learning, 8(3-4), 231-357, (2015).
- [19] Candes E. J., *Compressive sampling*, In Proceedings of the international congress of mathematicians, 3, 1433-1452, (2006).

- [20] Candes E. J. and Recht B., *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, 9(6), 717-772, (2009).
- [21] Candes E. J. and Romberg J. K., *l1-MAGIC: Recovery of Sparse Signals via Convex Programming*, A Collection of MATLAB Routines for Solving the Convex Optimization Programs Central to Compressive Sensing, 19 p., (2005), <https://statweb.stanford.edu/~candes/l1magic/> .
- [22] Candes E. J., Romberg J. K. and Tao T., *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 59(8), 1207-1223, (2006).
- [23] Cauchy A., *Méthode générale pour la résolution des systemes d'équations simultanées*, Comp. Rend. Sci. Paris, 536-538, (1847).
- [24] Chang C. C. and Lin C. J., *LIBSVM: A library for support vector machines*, ACM transactions on intelligent systems and technology (TIST), 2(3), 1-27, (2011), <http://www.csie.ntu.edu.tw/~cjlin/libsvm> .
- [25] Chen S. S., Donoho D. L. and Saunders M. A., *Atomic decomposition by basis pursuit*, SIAM Review, 43(1), 129-159, (2001).
- [26] Combettes P. L. and Pesquet J. C., *Proximal splitting methods in signal processing*, Fixed-point algorithms for inverse problems in science and engineering, Springer, New York, 185-212, (2011).
- [27] Defazio A., Bach F. and Lacoste-Julien S., *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, In Advances in Neural Information Processing Systems, 1646-1654, (2014).
- [28] Donoho D. L., *Compressed sensing*, IEEE Transactions on Information Theory, 52(4), 1289-1306, (2006).
- [29] Donoho D. L., *Neighborly polytopes and sparse solutions of underdetermined linear equations*, Technical Report, Stanford University, (2005).

- [30] Donoho D. L. and Tsaig Y., *Fast solution of l_1 -norm minimization problems when the solution may be sparse*, IEEE Transactions on Information Theory, 54(11), 4789-4812, (2008).
- [31] Donoho D. L., Elad M. and Temlyakov V. N., *Stable recovery of sparse overcomplete representations in the presence of noise*, IEEE Transactions on Information Theory, 52(1), 6-18, (2006).
- [32] Efron B., Hastie T., Johnstone I. and Tibshirani R., *Least angle regression*, The Annals of Statistics, 32(2), 407-499, (2004).
- [33] Fiacco A. V. and McCormick G. P., *The sequential unconstrained minimization technique for nonlinear programming, a primal-dual method*, Management Science, 10(2), 360-366, (1964).
- [34] Fiacco A. V. and McCormick G. P., *Nonlinear programming: sequential unconstrained minimization techniques*, SIAM, 210 p., (1990).
- [35] Figueiredo M. A., Nowak R. D. and Wright S. J., *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, 1(4), 586-597, (2007).
- [36] Fountoulakis K. and Gondzio J., *Performance of First-and Second-Order Methods for Big Data Optimization*, Technical Report ERGO-15-005, University of Edinburgh, (2015).
- [37] Fountoulakis K. and Gondzio J., *A second-order method for strongly convex l_1 -regularization problems*, Mathematical Programming, 156(1-2), 189-219, (2016).
- [38] Fountoulakis K., Gondzio J. and Zhlobich P., *Matrix-free interior point method for compressed sensing problems*, Mathematical Programming Computation, 6(1), 1-31, (2014).
- [39] Friedman J., Hastie T. and Tibshirani R., *Sparse inverse covariance estimation with the graphical lasso*, Biostatistics, 9(3), 432-441, (2008).

- [40] Genkin A., Lewis D. D. and Madigan D., *Large-scale Bayesian logistic regression for text categorization*, Technometrics, 49(3), 291-304, (2007).
- [41] Ghadimi E., Teixeira A., Shames I. and Johansson M., *On the optimal step-size selection for the alternating direction method of multipliers*, In Preprints of the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems, International Federation of Automatic Control, Santa Barbara, 139-144, (2012).
- [42] Ghadimi E., Teixeira A., Shames I. and Johansson M., *Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems*, IEEE Transactions on Automatic Control, 60(3), 644-658, (2014).
- [43] Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., Coller H., Loh M. L., Downing J. R., Caligiuri M. A., Bloomfield C. D., and Lander E. S., *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*, Science, 286(5439), 531-537, (1999).
- [44] Gondzio J., *Interior point methods 25 years later*, European Journal of Operational Research, 218(3), 587-601, (2012).
- [45] Gondzio J., *Matrix-free interior point method*, Computational Optimization and Applications, 51(2), 457-480, (2012).
- [46] Hale E. T., Yin W. and Zhang Y., *Fixed-point continuation for l_1 -minimization: Methodology and convergence*, SIAM Journal on Optimization, 19(3), 1107-1130, (2008).
- [47] Halická M., *Dvadsať rokov moderných metód vnútorného bodu*, Pokroky matematiky, fyziky a astronomie, 49(3), 234-244, (2004).
- [48] Hamala M. and Trnovská M., *Nelineárne programovanie, teória a algoritmy*, Epos, Bratislava, 339 p., (2013).
- [49] Hansen P. C., Nagy J. G. and O'leary D. P., *Deblurring images: matrices, spectra, and filtering*, SIAM, 130 p., (2006).

- [50] Hastie T., Rosset S., Tibshirani R. and Zhu J., *The entire regularization path for the support vector machine*, Journal of Machine Learning Research, 5(Oct), 1391-1415, (2004).
- [51] Hosmer Jr. D. W., Lemeshow S. and Sturdivant R. X., *Applied logistic regression*, Wiley Series in Probability and Statistics, John Wiley and Sons, 528 p., (2013).
- [52] Kanno Y., *A fast first-order optimization approach to elastoplastic analysis of skeletal structures*, Optimization and Engineering, 17(4), 861-896, (2016).
- [53] Karmarkar N., *A new polynomial-time algorithm for linear programming*, Combinatorica, 4(4), 373-395, (1984).
- [54] Keerthi S. S. and DeCoste D., *A modified finite Newton method for fast solution of large scale linear SVMs*, Journal of Machine Learning Research, 6(Mar), 341-361, (2005).
- [55] Kim S. J., Koh K., Boyd S. and Gorinevsky D., l_1 *Trend Filtering*, SIAM Review, 51(2), 339-360, (2009).
- [56] Kim S. J., Koh K., Lustig M., Boyd S. and Gorinevsky D., *An Interior-Point Method for Large-Scale l_1 -Regularized Least Squares*, IEEE Journal of Selected Topics in Signal Processing, 1(4), 606-617, (2007).
- [57] Kottnerus J. A., *Application of logistic regression to the analysis of diagnostic data: exact modeling of a probability tree of multiple binary variables*, Medical Decision Making, 12(2), 93-108, (1992).
- [58] Koh K., Kim S. J. and Boyd S., *An interior-point method for large-scale l_1 -regularized logistic regression*, Journal of Machine Learning Research, 8(Jul), 1519-1555, (2007).
- [59] Kojima M., Megiddo N. and Mizuno S., *A primal-dual infeasible-interior-point algorithm for linear programming*, Mathematical Programming, 61(3), 263-280, (1993).

- [60] Konečný J. and Richtárik P., *Semi-stochastic gradient descent methods*, arXiv preprint arXiv:1312.1666, (2013).
- [61] Kukumberg R., *Dva prístupy na riešenie úloh optimálneho topologického dizajnu*, Študentská vedecká konferencia FMFI UK, Bratislava 2014 : Zborník príspevkov, FMFI UK, Bratislava, 6-16, (2014).
- [62] Kukumberg R., *Proximal-gradient metóda konvexného programovania*, Bakalárska práca, FMFI UK, Bratislava, (2011).
- [63] Kukumberg R., *Najmenšie štvorce regulované l_1 normou - metódy a aplikácie*, Diplomová práca, FMFI UK, Bratislava, (2013).
- [64] Kukumberg R., *Two approaches for solving l_1 -regularized least squares with application to truss topology design*, Acta Mathematica Universitatis Comenianae, 84(2), 297-308, (2015).
- [65] Lee J. D., Recht B., Srebro N., Tropp J. and Salakhutdinov R. R., *Practical large-scale optimization for max-norm regularization*, In Advances in Neural Information Processing Systems, 1297-1305, (2010).
- [66] Lee J. D., Sun Y. and Saunders M. A., *Proximal Newton-type methods for minimizing composite functions*, SIAM Journal on Optimization, 24(3), 1420-1443, (2014).
- [67] Lee S. I., Lee H., Abbeel P. and Ng A. Y., *Efficient l_1 regularized logistic regression*, In AAAI, 6, 401-408, (2006).
- [68] Lemeshow S. and Hosmer Jr. D. W., *Logistic regression analysis: applications to ophthalmic research*, American Journal of Ophthalmology, 147(5), 766-767, (2009).
- [69] Lewis D. D., Yang Y., Rose T. G. and Li F., *Rcv1: A new benchmark collection for text categorization research*, Journal of machine learning research, 5(Apr), 361-397, (2004).

- [70] Liao J. G. and Chin K. V., *Logistic regression for disease classification using microarray data: model selection in a large p and small n case*, Bioinformatics, 23(15), 1945-1951, (2007).
- [71] Lorenz D. A., *Constructing test instances for basis pursuit denoising*, IEEE Transactions on Signal Processing, 61(5), 1210-1214, (2012).
- [72] Lorenz D. A. and Tran-Dinh Q., *Non-stationary Douglas–Rachford and alternating direction method of multipliers: adaptive step-sizes and convergence*, Computational Optimization and Applications, 74(1), 67-92, (2019).
- [73] Lustig M., Donoho D. and Pauly J. M., *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 58(6), 1182-1195, (2007).
- [74] Ma J., Saul L. K., Savage S. and Voelker G. M., *Identifying suspicious URLs: an application of large-scale online learning*, In Proceedings of the 26th annual international conference on machine learning (ICML), 681-688, (2009).
- [75] Mairal J., *Incremental majorization-minimization optimization with application to large-scale machine learning*, SIAM Journal on Optimization, 25(2), 829-855, (2015).
- [76] Mairal J., *Optimization with first-order surrogate functions*, In International Conference on Machine Learning, 783-791, (2013).
- [77] Nemirovski A., Juditsky A., Lan G. and Shapiro A., *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization, 19(4), 1574-1609 (2009).
- [78] Nesterov Y., *Gradient methods for minimizing composite functions*, Mathematical Programming, 140(1), 125-161, (2013).
- [79] Nesterov Y., *Introductory Lectures on Convex Programming. Volume I: Basic course*, Lecture notes, 212 p., (1998).

- [80] Nesterov Y. and Nemirovskii A., *Interior-point polynomial algorithms in convex programming*, SIAM, Philadelphia, 405 p., (1994).
- [81] Nocedal J. and Wright S., *Numerical optimization*, Springer Science & Business Media, (2006).
- [82] Parikh N. and Boyd S., *Proximal algorithms*, Foundations and Trends in Optimization, 1(3), 127-239, (2014).
- [83] Parra L. C., Spence C. D., Gerson A. D. and Sajda P., *Recipes for the linear analysis of EEG*, Neuroimage, 28(2), 326-341, (2005).
- [84] Polyak B. T., *Introduction to optimization*, Translations Series in Mathematics and Engineering, Optimization Software, New York, 438 p., (1987).
- [85] Richtárik P. and Takáč M., *Efficient serial and parallel coordinate descent methods for huge-scale truss topology design*, In Operations Research Proceedings, Springer, 27-32, (2012).
- [86] Richtárik P. and Takáč M., *Parallel coordinate descent methods for big data optimization*, Mathematical Programming, 156(1-2), 433-484, (2016).
- [87] Roth V., *The generalized LASSO*, IEEE Transactions on Neural Networks, 15(1), 16-28, (2004).
- [88] Sardy S., Bruce A. G. and Tseng P., *Block coordinate relaxation methods for non-parametric wavelet denoising*, Journal of Computational and Graphical Statistics, 9(2), 361-379, (2000).
- [89] Saunders M. A., Kim B., Maes C., Akle S. and Zahr M., *PDCO: Primal-dual interior method for convex objectives*, (2002), Software available at <http://www.stanford.edu/group/SOL/software/pdco.html> .
- [90] Schmidt M., *Graphical model structure learning using l_1 -regularization*, PhD Thesis, University of British Columbia, (2010).
- [91] Schmidt M., Fung G. and Rosales R., *Optimization methods for l_1 -regularization*, Technical Report TR-2009-19, University of British Columbia, 20 p., (2009).

- [92] Schmidt M., Le Roux N. and Bach F., *Minimizing Finite Sums with the Stochastic Average Gradient*, arXiv preprint arXiv:1309.2388, (2013).
- [93] Shalev-Shwartz S. and Ben-David S., *Understanding machine learning: From theory to algorithms*, Cambridge University Press, (2014).
- [94] Shi J., Yin W., Osher S. and Sajda P., *A fast hybrid algorithm for large-scale l_1 -regularized logistic regression*, Journal of Machine Learning Research, 11(Feb), 713-741, (2010).
- [95] Steyerberg E. W., Eijkemans M. J. C. and Habbema J. D. F., *Application of shrinkage techniques in logistic regression analysis: a case study*, Statistica Neerlandica, 55(1), 76-88, (2001).
- [96] Teo C. H., Smola A., Vishwanathan S. V. N. and Le Q. V., *A scalable modular convex solver for regularized risk minimization*, In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 727-736, (2007).
- [97] Tibshirani R., *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society, Series B, 58(1), 267-288, (1996).
- [98] Toh K. C. and Yun S., *An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems*, Pacific Journal of Optimization, 6(3), 615-640, (2010).
- [99] Trofimov I. and Genkin A., *Distributed coordinate descent for l_1 -regularized logistic regression*, In International Conference on Analysis of Images, Social Networks and Texts, Springer International Publishing, 243-254, (2015).
- [100] Tsuruoka Y., McNaught J., Tsujii J.I.C. and Ananiadou S., *Learning string similarity measures for gene/protein name dictionary look-up using logistic regression*, Bioinformatics, 23(20), 2768-2774, (2007).
- [101] Vanderberghe L., *Optimization methods for large-scale systems*, Lecture notes, UCLA EE 236C, (2010).

- [102] Vattikuti S., Lee J. J., Chang C. C., Hsu S. D. and Chow C. C., *Applying compressed sensing to genome-wide association studies*, GigaScience, 3(1), 17 p., (2014).
- [103] Webb S., Caverlee J. and Pu C., *Introducing the Webb Spam Corpus: Using Email Spam to Identify Web Spam Automatically*, CEAS, (2006).
- [104] Wen Z., Yin W., Goldfarb D. and Zhang Y., *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*, SIAM Journal on Scientific Computing, 32(4), 1832-1857, (2010).
- [105] Wright S. J., *Primal-dual interior-point methods*, SIAM, Philadelphia, 289 p., (1997).
- [106] Wright S. J., Nowak R. D. and Figueiredo M. A., *Sparse reconstruction by separable approximation*, IEEE Transactions on Signal Processing, 57(7), 2479-2493, (2009).
- [107] Xiao L. and Zhang T., *A proximal-gradient homotopy method for the sparse least-squares problem*, SIAM Journal on Optimization, 23(2), 1062-1091, (2013).
- [108] Yuan G. X., Chang K. W., Hsieh C. J. and Lin C. J., *A comparison of optimization methods and software for large-scale l_1 -regularized linear classification*, Journal of Machine Learning Research, 11, 3183-3234, (2010).
- [109] Zhang P. G. C., *A fast dual projected newton method for l_1 -regularized least squares*, Tsinghua University, Beijing, (2011).

Dodatok A

Prílohy

A.1 Vlastnosti funkcií

Pri proximálnych metódach, resp. pri samotnom definovaní proximálneho operátora požadujeme, aby funkcia g bola tzv. uzavretá vlastná konvexná funkcia (angl. closed proper convex function).

Definícia A.1.1. (uzavretá vlastná konvexná funkcia)[82] Funkcia $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ sa nazýva uzavretá vlastná konvexná funkcia, ak jej epigraf

$$\text{epi } g = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} | g(x) \leq t\} \quad (\text{A.1})$$

je neprázdna uzavretá konvexná množina.

Pre konvergenciu proximálnych metód je často požadovaný predpoklad Lipschitzovskosti gradientu funkcie f . Lipschitzovská konštanta L sa takisto používa pri voľbe konštantných dĺžok kroku pri týchto metódach.

Definícia A.1.2. Funkcia n premenných $f(x)$ má Lipschitzovsky spojitý gradient s konštantou $L > 0$, ak pre každú dvojicu bodov $x, y \in \mathbb{R}^n$ platí:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|. \quad (\text{A.2})$$

Poznámka A.1.1. V literatúre sa funkcia s L -Lipschitzovsky spojitým gradientom nazýva taktiež L -hladká.

Tvrdenie A.1.1. [79] Dvakrát spojitá diferencovateľná funkcia na $\mathbb{R}^n$ má Lipschitzovský spojitý gradient s konštantou $L > 0$ práve vtedy, keď

$$\|\nabla^2 f(x)\| \leq L \quad \forall x \in \mathbb{R}^n. \quad (\text{A.3})$$

Veta A.1.1. [79] Nech funkcia $f(x)$ má Lipschitzovský spojitý gradient s konštantou $L > 0$. Potom pre každú dvojicu bodov $x, y \in \mathbb{R}^n$ platí:

$$|f(y) - f(x) - \nabla f(x)^T(y - x)| \leq \frac{L}{2} \|y - x\|^2. \quad (\text{A.4})$$

Vďaka predchádzajúcej vete je možné funkciu $f(x)$ s Lipschitzovským spojitým gradientom vo fixnom bode $x_f \in \mathbb{R}^n$ ohraničiť dvomi kvadratickými funkciami zdola aj zhora.

Definícia A.1.3. [79] Spojitá diferencovateľná funkcia $f(x)$ sa nazýva silnokonvexná na $\mathbb{R}^n$, ak existuje taká kladná konštanta $\mu > 0$, že pre každú dvojicu bodov $x, y \in \mathbb{R}^n$ platí:

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2} \|y - x\|^2. \quad (\text{A.5})$$

Z predchádzajúcej definície priamo vyplýva (ak $x = x^*$), že minimum silnokonvexnej funkcie je jednoznačné.

A.2 Subgradient funkcie

Proximálny operátor úzko súvisí s tzv. subgradientami (resp. subdiferenciálom) funkcie g (o čom nám hovorí Veta 2.1.1.).

Definícia A.2.1. (Subgradient)[15] Vektor $v \in \mathbb{R}^n$ nazývame subgradientom funkcie $g : \mathbb{R}^n \rightarrow \mathbb{R}$ v bode $x \in \text{dom } g$, ak pre všetky body $z \in \text{dom } g$ spĺňa

$$g(z) \geq g(x) + v^T(z - x). \quad (\text{A.6})$$

Poznámka A.2.1. Ak je g konvexná diferencovateľná funkcia, tak potom je jej gradient v bode x rovný subgradientu (vidno z podmienky konvexnosti prvého rádu). Subgradient však môže existovať aj pre funkciu nediferencovateľnú v bode x . V bode nediferencovateľnosti x môže existovať viac subgradientov funkcie g v bode x (napríklad pre funkciu $g(x) = |x|$ v bode $x = 0$).

Definícia A.2.2. (Subdiferenciál)[15] Množina subgradientov funkcie g v bode x sa nazýva subdiferenciál funkcie g v bode x a značí sa $\partial g(x)$.

Veta A.2.1. (Minimum nediferencovateľnej funkcie)[15] Bod x^* je bodom minima konvernej funkcie g práve vtedy, keď g je subdiferencovateľná v bode x^* a platí

$$0 \in \partial g(x^*), \quad (\text{A.7})$$

t.j. $v = 0$ je subgradientom funkcie g v bode x^* .

Dôkaz vychádza priamo z faktu: $g(x) \geq g(x^*)$ pre všetky x z definičného oboru funkcie g .

Podmienka $0 \in \partial g(x^*)$ sa pre funkciu g diferencovateľnú v bode x redukuje na známu podmienku $0 = \nabla g(x^*)$.

A.3 Rýchlosť konverencie

V tejto práci častokrát spomíname rôzne rýchlosti konverencie. Podľa [79] rýchlosť konverencie je:

- tzv. *sublineárna*, ak je popísaná mocninovou funkciou čísla iterácie.

Napríklad $r_k \leq \frac{c}{k}$.

- tzv. *lineárna*, ak je daná exponenciálnou funkciou čísla iterácie.

Napríklad $r_k \leq c(1 - q)^k$.

- tzv. *kvadratická*, ak má tvar dvojnásobnej exponenciálnej funkcie čísla iterácie.

Napríklad $r_{k+1} \leq cr_k^2 \leq \dots \leq \frac{1}{c}(cr_0)^{2^k}$.

A.4 Sherman–Morrison–Woodbury formula

Podľa [48] Sherman–Morrison–Woodbury formula vyzerá nasledovne

$$(A - US^{-1}V^T)^{-1} = A^{-1} + A^{-1}U(S - V^T A^{-1}U)^{-1}V^T A^{-1}, \quad (\text{A.8})$$

kde matice U a V sú typu $n \times m$ a matice $A \in \mathbb{R}^{n \times n}$ a $S \in \mathbb{R}^{m \times m}$ sú regulárne.

Dodatok B

Tabulky

V tejto časti uvádzame podrobné výsledky experimentov spracované v tabulkách.

Tabuľka B.1: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.2) s hodnotou regularizačného parametra $\lambda = 0.8\lambda_{max}$.

Metóda	Presnosť ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		pr		max		pr		max		pr		max		pr		max		pr	
	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max
ADMM	0,43	0,31	0,69	18	1	46	0,91	0,72	1,15	91	63	120	1,41	1,14	1,68	166	127	208						
APG	0,02	0,00	0,05	15	1	33	0,22	0,10	0,32	154	67	224	1,17	0,20	1,56	826	137	1093						
APGls	0,02	0,00	0,03	5	1	11	0,12	0,07	0,20	48	25	74	0,49	0,19	0,73	194	73	266						
DPG1	0,04	0,00	0,09	15	1	33	0,40	0,17	0,65	160	67	263	2,30	0,82	2,98	941	338	1221						
DPG1ls	0,01	0,00	0,03	4	1	9	0,07	0,05	0,07	24	16	28	0,10	0,09	0,11	37	33	39						
DPG2	0,04	0,00	0,09	15	1	33	0,25	0,17	0,39	101	67	158	0,41	0,26	0,56	164	103	224						
DPG2ls	0,01	0,00	0,03	4	1	9	0,06	0,05	0,07	21	16	26	0,08	0,06	0,10	29	22	38						
DPG3	0,04	0,00	0,09	15	1	33	0,29	0,17	0,41	116	67	165	0,52	0,31	0,68	211	126	276						
DPG3ls	0,01	0,00	0,03	4	1	9	0,06	0,05	0,07	22	16	26	0,08	0,07	0,10	30	24	38						
PD11i	0,17	0,12	0,27	11	8	16	0,29	0,21	0,46	18	15	24	0,40	0,29	0,59	25	22	32						
PD11ir	0,18	0,13	0,28	11	8	16	0,19	0,13	0,28	11	9	16	0,19	0,13	0,28	15	10	21						
PD12i	0,26	0,18	0,36	10	8	14	0,44	0,30	0,58	16	15	20	0,59	0,40	0,77	23	21	27						
PD13	5,00	3,91	7,92	10	8	16	8,07	6,87	10,87	16	14	22	11,33	10,23	13,83	23	21	28						
PD21i	0,44	0,30	0,71	11	9	17	0,74	0,47	1,05	18	15	23	0,98	0,61	1,39	25	22	32						
PD22i	0,63	0,42	0,94	12	9	17	1,01	0,70	1,41	18	15	24	1,31	0,92	1,78	24	21	30						
PD23	9,96	8,23	14,50	11	9	16	15,53	13,58	19,90	17	15	22	21,28	18,98	26,19	23	21	29						
PGls	0,01	0,00	0,03	6	1	13	0,06	0,05	0,06	31	29	33	0,07	0,06	0,07	38	36	39						

Tabuľka B.2: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.2) s hodnotou regularizačného parametra $\lambda = 0.5\lambda_{max}$.

Presnosť ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácii			čas			# iterácii			čas			# iterácii		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
		0,48	0,44	0,59	26	21	44	0,95	0,85	1,07	96	83	116	1,50	1,35	1,75	180	158	218
	ADMM	0,08	0,07	0,10	57	49	69	0,51	0,41	0,66	359	287	467	2,17	1,76	2,61	1536	1246	1844
	APG	0,05	0,04	0,08	21	17	29	0,26	0,21	0,38	104	81	151	0,88	0,72	1,43	343	288	544
	APGls	0,15	0,12	0,18	57	49	69	0,90	0,72	1,13	367	294	465	3,84	3,31	4,65	1577	1360	1916
	DPG1	0,04	0,04	0,05	14	13	18	0,09	0,08	0,12	33	29	44	0,15	0,11	0,24	56	40	89
	DPG1ls	0,15	0,12	0,19	57	49	69	0,41	0,29	0,95	163	117	383	0,64	0,52	1,15	257	209	461
	DPG2	0,04	0,04	0,05	14	13	18	0,08	0,06	0,10	28	23	37	0,10	0,09	0,15	37	31	54
	DPG2ls	0,14	0,12	0,18	57	49	69	0,46	0,35	0,99	185	143	399	0,80	0,60	1,27	327	245	513
	DPG3	0,04	0,04	0,05	14	13	18	0,08	0,07	0,10	29	25	38	0,11	0,09	0,15	39	33	56
	DPG3ls	0,19	0,15	0,30	10	8	14	0,43	0,35	0,54	16	15	20	0,59	0,49	0,73	23	21	27
	PD11i	0,21	0,16	0,31	10	8	14	0,25	0,20	0,36	11	10	16	0,25	0,20	0,36	15	13	20
	PD11lr	0,33	0,27	0,45	9	8	13	0,67	0,55	0,79	16	15	19	0,90	0,72	1,12	22	21	26
	PD12i	4,42	3,94	6,42	9	8	13	7,62	6,90	9,85	15	14	20	10,78	10,33	12,81	22	21	26
	PD13	0,52	0,44	0,74	10	9	15	1,06	0,84	1,36	17	15	22	1,43	1,15	1,76	24	21	28
	PD21i	0,76	0,63	1,06	10	9	14	1,47	1,20	1,79	16	15	20	1,96	1,60	2,42	23	21	27
	PD22i	8,93	7,42	12,98	10	8	14	14,60	13,60	19,31	16	15	21	20,28	18,98	24,67	22	21	27
	PD23	0,04	0,03	0,05	23	19	28	0,07	0,06	0,09	37	35	48	0,09	0,07	0,14	49	40	74
	PGls																		

Tabuľka B.3: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.2) s hodnotou regularizačného parametra $\lambda = 0.2\lambda_{max}$.

Metóda	Presnosť ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM		0,42	0,39	0,64	17	13	52	0,62	0,51	1,10	46	32	121	0,84	0,66	1,54	79	54	188
APG		0,17	0,12	0,22	116	85	147	1,11	0,75	1,42	788	531	1011	5,00	3,25	6,62	3554	2308	4718
APGls		0,23	0,14	0,39	89	54	154	1,53	0,81	2,88	578	320	1065	6,75	3,32	14,01	2479	1239	5119
DPG1		0,29	0,21	0,37	116	85	147	1,92	1,30	2,46	788	533	1014	8,70	5,72	11,57	3586	2351	4781
DPG1ls		0,11	0,08	0,15	39	30	51	0,43	0,24	0,64	160	90	226	1,22	0,50	1,93	446	192	700
DPG2		0,29	0,21	0,36	116	85	147	1,82	0,95	2,37	744	388	971	2,55	1,34	4,02	1042	546	1647
DPG2ls		0,11	0,08	0,14	39	30	51	0,24	0,17	0,36	88	60	131	0,34	0,22	0,49	124	77	180
DPG3		0,29	0,21	0,37	116	85	147	1,83	1,01	2,36	752	416	971	2,71	1,48	4,09	1115	607	1678
DPG3ls		0,11	0,08	0,14	39	30	51	0,24	0,15	0,38	90	57	134	0,37	0,23	0,62	138	84	236
PD1li		0,28	0,23	0,35	10	10	11	0,81	0,57	0,99	16	16	17	1,22	0,83	1,57	23	22	24
PD1lir		0,29	0,25	0,36	10	10	11	0,34	0,25	0,42	13	12	14	0,36	0,26	0,44	17	16	18
PD12i		0,51	0,40	0,59	10	10	11	1,25	0,93	1,55	16	16	17	1,84	1,31	2,41	23	22	23
PD13		5,38	4,92	5,95	11	10	12	8,52	8,34	8,96	17	17	18	11,69	11,29	12,31	24	23	25
PD21i		0,74	0,60	0,84	10	10	10	1,96	1,44	2,52	16	16	17	2,92	1,98	3,70	23	22	23
PD22i		1,23	0,92	1,46	10	10	11	2,80	2,12	3,48	16	16	17	4,05	2,83	5,14	23	22	24
PD23		9,22	8,26	10,20	10	9	11	14,87	14,42	15,56	16	16	17	20,75	19,85	21,73	23	22	24
PGls		0,18	0,09	0,30	102	51	169	0,50	0,18	0,88	287	104	508	0,88	0,27	1,67	504	153	965

Tabuľka B.4: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.2) s hodnotou regularizačného parametra $\lambda = 0.1\lambda_{max}$.

Metóda	Presnosť ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM		0,49	0,47	0,58	27	25	35	0,81	0,70	0,97	76	59	102	1,21	1,04	1,43	136	112	171
APG		0,28	0,19	0,35	199	137	244	1,88	1,30	2,32	1336	923	1647	9,56	6,18	12,27	6815	4421	8703
APGls		0,49	0,34	0,69	193	134	256	3,52	2,35	4,73	1301	874	1741	18,05	11,33	25,67	6600	4143	9345
DPG1		0,49	0,35	0,61	199	137	244	3,24	2,23	4,01	1334	924	1643	16,65	10,68	21,30	6881	4459	8751
DPG1ls		0,22	0,13	0,29	81	50	108	1,06	0,62	1,42	394	221	522	4,00	1,94	5,87	1417	676	2057
DPG2		0,49	0,34	0,63	199	137	244	2,99	1,84	3,90	1229	758	1605	4,12	2,46	6,85	1691	1012	2824
DPG2ls		0,22	0,13	0,28	81	50	108	0,51	0,30	0,95	193	111	364	0,76	0,43	1,27	285	159	471
DPG3		0,49	0,33	0,63	199	137	244	3,01	1,89	3,93	1243	788	1605	4,58	3,02	6,87	1889	1244	2842
DPG3ls		0,22	0,13	0,28	81	50	108	0,53	0,30	0,99	201	113	375	0,93	0,47	1,52	348	177	558
PD11i		0,40	0,30	0,58	11	11	12	1,38	0,93	1,76	18	17	18	2,18	1,46	2,72	24	24	25
PD11ir		0,41	0,32	0,60	11	11	12	0,49	0,36	0,65	15	14	16	0,54	0,40	0,70	19	18	20
PD12i		0,74	0,57	0,96	11	11	12	2,06	1,46	2,56	18	17	18	3,22	2,20	3,94	24	24	25
PD13		6,10	5,88	6,53	12	12	13	9,29	8,86	9,56	19	18	19	12,39	11,82	12,86	25	24	26
PD21i		1,09	0,86	1,37	11	11	11	3,27	2,34	4,20	17	17	18	5,29	3,54	6,61	24	23	25
PD22i		1,90	1,46	2,31	11	11	12	4,59	3,38	5,63	17	17	18	7,04	4,91	8,67	24	24	25
PD23		10,31	9,98	11,01	11	11	12	16,14	15,31	16,54	18	17	18	21,75	20,73	22,70	24	23	25
PGLs		0,68	0,28	1,08	392	157	612	2,31	0,81	3,68	1339	475	2123	4,40	1,54	7,36	2555	888	4293

Tabuľka B.6: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.2) s hodnotou regularizačného parametra $\lambda = 0.01\lambda_{max}$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácii		čas		čas		# iterácii		čas		# iterácii		čas		# iterácii	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,66	0,60	0,75	53	45	60	1,84	1,36	2,52	231	161	341	4,51	2,40	7,86	638	319	1149
APG	1,26	0,96	1,47	903	687	1059	7,51	5,93	8,74	5411	4250	6289	47,74	37,64	58,06	34433	27268	41699
DPG1ls	1,29	0,96	1,51	473	355	557	6,27	4,85	7,42	2242	1728	2653	35,27	26,76	43,61	12547	9521	15525
DPG2	2,20	1,68	2,57	903	687	1059	9,49	7,28	11,34	3917	3005	4677	16,97	11,62	23,18	7010	4808	9561
DPG2ls	1,32	0,96	1,62	473	355	557	4,04	2,76	5,33	1413	962	1836	7,53	5,04	10,22	2616	1742	3528
DPG3	2,20	1,67	2,57	903	687	1059	8,89	5,54	10,98	3667	2279	4532	20,06	13,47	28,28	8283	5547	11634
DPG3ls	1,29	0,93	1,54	473	355	557	4,01	2,65	5,10	1415	947	1788	8,69	6,07	13,82	3033	2123	4832
PD11i	1,71	1,20	2,18	15	14	16	9,28	6,16	12,30	22	22	23	17,62	10,80	25,64	29	28	30
PD11ir	1,48	1,16	1,81	15	14	16	2,87	2,09	3,55	20	19	21	3,76	2,72	4,76	24	23	25
PD12i	3,01	2,37	3,75	15	15	16	11,98	8,10	16,00	23	21	24	21,75	14,22	29,98	29	28	30
PD13	8,08	7,38	8,52	16	15	17	11,56	11,29	11,94	23	23	24	14,80	14,26	16,81	30	29	33
PD21i	4,46			14			27,18			22			47,05			28		
PD22i	9,67			15			31,56			22			52,62			28		
PD23	14,17	12,74	14,72	16	14	16	20,47	19,90	21,79	23	22	24	26,40	25,35	27,32	29	28	30
PGLs	19,27			11K			160			91K			533			0,3M		

Tabuľka B.7: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.2) s hodnotou regularizačného parametra $\lambda = 0.005\lambda_{max}$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,79	0,72	0,95	70	59	78	3,13	2,18	5,21	412	271	745	10,65	5,44	20,86	1515	723	2939
APG	1,84	1,42	2,16	1319	1022	1548	10,56	8,51	12,36	7583	6132	8901	67,06	54,30	77,24	48141	38967	55688
DPG1ls	1,96	1,44	2,34	715	538	854	9,09	7,21	10,77	3237	2574	3833	51,15	40,47	61,50	18108	14322	21733
DPG2	3,23	2,50	3,85	1319	1022	1548	13,93	10,10	17,38	5713	4139	7157	29,21	18,54	41,02	11997	7598	16784
DPG2ls	2,03	1,49	2,44	719	538	854	6,63	4,79	8,50	2300	1674	2927	13,61	8,78	19,00	4697	3020	6557
DPG3	3,23	2,51	3,82	1319	1022	1548	13,76	9,61	17,98	5635	3933	7350	30,56	20,96	42,83	12536	8645	17541
DPG3ls	2,03	1,48	2,51	726	538	890	6,71	4,78	9,21	2341	1652	3220	14,12	9,53	19,87	4899	3328	6916
PD11i	2,55	1,77	3,40	16	15	17	16,67	10,78	24,17	24	23	25	33,50	19,67	49,16	30	29	31
PD11ir	2,28	1,74	2,78	16	15	17	5,36	3,87	7,11	22	21	23	7,14	5,21	10,27	26	25	27
PD12i	4,30	3,26	5,27	16	16	17	20,26	12,95	28,31	24	23	25	39,00	24,53	54,47	31	30	32
PD13	8,24	7,85	8,90	17	16	18	12,01	11,36	12,43	24	23	25	15,69	14,69	18,12	32	30	36
PD21i	7,99			16			50,62			24			94,77			30		
PD22i	14,56			16			54,62			24			98,36			30		
PD23	15,84	14,42	17,09	16	15	17	23,32	21,89	24,67	24	23	25	30,72	28,49	36,41	31	30	36

Tabuľka B.8: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 600$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácii		čas		čas		# iterácii		čas		čas		# iterácii		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,06	0,06	0,07	48	42	53	0,13	0,11	0,15	139	119	173	0,21	0,17	0,28	253	204	360
	APG	0,11	0,09	0,13	268	208	319	0,72	0,55	0,91	1697	1303	2101	4,05	2,95	5,18	9467	6887	11871
	APGs	0,19	0,12	0,31	253	150	403	1,18	0,66	1,95	1596	882	2630	6,53	3,31	11,70	8863	4543	15956
	DPG1	0,19	0,14	0,22	268	208	319	1,18	0,92	1,48	1695	1312	2104	6,82	4,97	8,96	9731	7110	12728
	DPG1s	0,06	0,04	0,08	73	56	89	0,25	0,17	0,33	311	224	398	0,93	0,59	1,30	1187	749	1651
	DPG2	0,19	0,14	0,22	268	208	319	0,80	0,50	1,33	1142	712	1885	1,19	0,79	1,83	1697	1130	2588
	DPG2s	0,06	0,05	0,08	73	56	89	0,15	0,09	0,19	176	115	232	0,23	0,15	0,30	276	181	374
	DPG3	0,19	0,14	0,22	268	208	319	0,79	0,50	1,33	1137	712	1893	1,39	0,89	1,93	1990	1284	2755
	DPG3s	0,06	0,04	0,08	73	56	89	0,15	0,09	0,20	181	122	238	0,26	0,17	0,36	327	226	443
	PD1i	0,26	0,19	0,36	15	14	16	0,62	0,47	0,81	22	21	22	0,92	0,65	1,20	28	27	29
	PD1ir	0,25	0,19	0,34	15	14	16	0,26	0,19	0,34	17	16	18	0,27	0,20	0,35	22	21	23
	PD12i	0,45	0,34	0,58	15	14	16	0,97	0,75	1,22	21	21	22	1,42	1,05	1,82	28	27	29
	PD13	0,66	0,62	0,74	16	15	17	0,94	0,89	1,00	22	21	23	1,21	1,16	1,25	28	28	29
	PD21i	0,64	0,49	0,80	16	15	17	1,47	1,11	1,90	23	22	24	2,13	1,52	2,76	29	28	30
	PD22i	0,97	0,73	1,23	15	15	16	2,01	1,52	2,45	22	21	23	2,88	2,10	3,80	28	27	29
	PD23	1,09	1,01	1,16	16	15	17	1,52	1,47	1,58	22	22	23	1,95	1,88	2,01	29	28	30
	PGls	0,15	0,09	0,21	296	177	434	0,58	0,28	1,03	1196	563	2098	1,21	0,55	2,14	2527	1137	4404

Tabuľka B.9: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 1200$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácii		čas		čas		# iterácii		čas		čas		# iterácii		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,15	0,14	0,19	34	30	38	0,28	0,24	0,32	95	80	104	0,42	0,37	0,47	167	142	184
	APG	0,15	0,12	0,18	227	173	271	1,01	0,79	1,24	1488	1181	1760	5,43	3,90	6,71	7907	5684	9878
	APGs	0,26	0,17	0,42	210	133	339	1,75	1,02	2,99	1365	809	2285	9,43	5,16	17,10	7173	3957	12907
	DPG1	0,26	0,20	0,31	227	173	271	1,71	1,35	2,01	1489	1181	1760	9,25	6,91	11,16	8050	5997	9851
	DPG1s	0,10	0,07	0,12	75	54	92	0,44	0,30	0,53	338	224	421	1,66	1,02	2,23	1230	759	1646
	DPG2	0,26	0,20	0,30	227	173	271	1,38	0,87	1,87	1209	765	1630	1,90	1,29	2,59	1663	1137	2279
	DPG2s	0,10	0,07	0,13	75	54	92	0,23	0,17	0,31	173	120	238	0,35	0,23	0,46	261	168	353
	DPG3	0,26	0,19	0,30	227	173	271	1,41	0,91	1,89	1237	803	1635	2,24	1,55	2,84	1961	1373	2482
	DPG3s	0,10	0,07	0,12	75	54	92	0,24	0,16	0,32	183	123	248	0,43	0,26	0,64	330	192	479
	PD1i	0,29	0,21	0,35	13	12	14	0,77	0,58	0,97	19	18	20	1,19	0,85	1,46	26	25	27
	PD1ir	0,29	0,20	0,34	13	12	14	0,31	0,22	0,37	16	15	17	0,33	0,23	0,39	20	19	21
	PD12i	0,49	0,36	0,59	13	12	13	1,18	0,92	1,46	19	18	20	1,78	1,27	2,15	26	25	26
	PD13	2,00	1,88	2,19	14	13	15	2,93	2,77	3,07	20	19	21	3,85	3,75	4,09	26	26	28
	PD21i	0,80	0,60	1,01	13	12	14	2,17	1,58	2,64	20	19	21	3,25	2,25	4,02	26	25	27
	PD22i	1,35	1,01	1,60	13	12	14	3,01	2,28	3,69	19	18	20	4,49	3,18	5,37	26	25	27
	PD23	3,55	3,42	3,98	13	13	15	5,30	5,01	5,57	20	19	21	6,96	6,84	7,20	26	26	27
	PGls	0,25	0,14	0,38	323	181	470	0,89	0,42	1,41	1164	543	1816	1,75	0,77	2,82	2299	1006	3678

Tabuľka B.10: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 2400$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácii			čas			# iterácii			čas			# iterácii		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,49	0,47	0,58	27	25	35	0,81	0,70	0,97	76	59	102	1,21	1,04	1,43	136	112	171
	APG	0,28	0,19	0,35	199	137	244	1,88	1,30	2,32	1336	923	1647	9,56	6,18	12,27	6815	4421	8703
	APGls	0,49	0,34	0,69	193	134	256	3,52	2,35	4,73	1301	874	1741	18,05	11,33	25,67	6600	4143	9345
	DPG1	0,49	0,35	0,61	199	137	244	3,24	2,23	4,01	1334	924	1643	16,65	10,68	21,30	6881	4459	8751
	DPG1ls	0,22	0,13	0,29	81	50	108	1,06	0,62	1,42	394	221	522	4,00	1,94	5,87	1417	676	2057
	DPG2	0,49	0,34	0,63	199	137	244	2,99	1,84	3,90	1229	758	1605	4,12	2,46	6,85	1691	1012	2824
	DPG2ls	0,22	0,13	0,28	81	50	108	0,51	0,30	0,95	193	111	364	0,76	0,43	1,27	285	159	471
	DPG3	0,49	0,33	0,63	199	137	244	3,01	1,89	3,93	1243	788	1605	4,58	3,02	6,87	1889	1244	2842
	DPG3ls	0,22	0,13	0,28	81	50	108	0,53	0,30	0,99	201	113	375	0,93	0,47	1,52	348	177	558
	PD1li	0,40	0,30	0,58	11	11	12	1,38	0,93	1,76	18	17	18	2,18	1,46	2,72	24	24	25
	PD1lir	0,41	0,32	0,60	11	11	12	0,49	0,36	0,65	15	14	16	0,54	0,40	0,70	19	18	20
	PD12i	0,74	0,57	0,96	11	11	12	2,06	1,46	2,56	18	17	18	3,22	2,20	3,94	24	24	25
	PD13	6,10	5,88	6,53	12	12	13	9,29	8,86	9,56	19	18	19	12,39	11,82	12,86	25	24	26
	PD2li	1,09	0,86	1,37	11	11	11	3,27	2,34	4,20	17	17	18	5,29	3,54	6,61	24	23	25
	PD22i	1,90	1,46	2,31	11	11	12	4,59	3,38	5,63	17	17	18	7,04	4,91	8,67	24	24	25
	PD23	10,31	9,98	11,01	11	11	12	16,14	15,31	16,54	18	17	18	21,75	20,73	22,70	24	23	25
	PGls	0,68	0,28	1,08	392	157	612	2,31	0,81	3,68	1339	475	2123	4,40	1,54	7,36	2555	888	4293

Tabuľka B.11: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 4800$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácii			čas			# iterácii			čas			# iterácii		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	1,95	1,66	2,84	21	17	67	2,82	2,40	5,15	60	45	172	3,98	3,16	7,48	111	80	278
	APG	0,54	0,35	0,68	165	106	207	3,66	2,35	4,63	1136	733	1441	17,25	9,75	22,88	5377	3044	7128
	APGls	0,94	0,77	1,59	162	134	275	6,39	5,19	11,37	1118	907	1988	30,15	23,42	59,47	5283	4095	10416
	DPG1	0,90	0,58	1,13	165	106	207	6,14	3,93	7,80	1136	734	1441	29,15	16,90	38,66	5404	3143	7157
	DPG1ls	0,49	0,31	0,67	79	50	110	2,46	1,19	3,55	407	200	588	8,51	3,10	13,41	1410	517	2222
	DPG2	0,90	0,58	1,13	165	106	207	5,98	3,11	7,78	1105	576	1441	9,38	3,92	15,75	1733	724	2900
	DPG2ls	0,49	0,31	0,65	79	50	110	1,31	0,50	2,31	214	78	373	1,84	0,75	3,11	297	118	509
	DPG3	0,90	0,58	1,13	165	106	207	6,00	3,25	7,83	1109	602	1441	9,85	4,17	15,76	1824	774	2914
	DPG3ls	0,49	0,31	0,69	79	50	110	1,42	0,60	2,43	235	98	401	2,14	0,91	3,82	354	148	629
	PD11i	0,61	0,51	0,71	10	9	11	2,30	1,48	2,95	17	16	17	3,76	2,33	5,18	23	22	24
	PD11ir	0,64	0,53	0,73	10	9	10	0,86	0,62	1,07	14	13	14	1,01	0,68	1,23	18	17	19
	PD12i	1,05	0,86	1,31	10	10	11	3,21	2,15	4,09	17	16	17	5,15	3,32	6,68	23	22	23
	PD13	20,72			11			32,17			17			43,55			23		
	PD21i	1,44	1,07	1,79	9	9	10	5,17	3,47	6,51	16	15	16	8,23	5,00	10,57	22	22	24
	PD22i	2,81	2,18	3,46	10	9	10	7,15	4,80	8,72	16	16	16	11,00	7,17	13,99	22	22	23
	PD23	33,51			10			56,30			17			76,07			23		
	PGls	1,55	0,46	2,71	401	121	705	4,93	1,23	9,45	1278	325	2448	9,01	2,06	17,38	2335	539	4504

Tabuľka B.12: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 9600$ a $n = 12000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	8,66	8,10	10,04	26	20	42	13,16	11,57	16,12	82	62	117	18,98	16,16	22,26	155	118	193
APG	1,08	0,75	1,35	144	100	178	7,30	4,96	9,04	981	666	1204	32,33	20,22	41,84	4351	2729	5599
APGls	1,98			153			13,98			1098			62,36			4908		
DPG1	1,45			114			9,79			771			40,90			3262		
DPG1ls	1,10	0,70	1,48	80	50	107	5,69	2,92	8,24	423	217	607	19,45	7,91	30,40	1454	590	2252
DPG2	1,81	1,25	2,24	144	100	178	12,24	8,28	15,06	981	666	1204	22,87	12,28	35,30	1832	984	2835
DPG2ls	1,09	0,68	1,48	79	46	107	3,38	1,26	8,22	247	88	607	4,41	1,83	9,43	322	129	690
DPG3	1,81	1,25	2,24	144	100	178	12,23	8,28	15,13	981	666	1204	23,36	12,78	35,91	1875	1028	2884
DPG3ls	1,10	0,70	1,47	80	50	107	3,64	1,43	8,21	270	105	607	5,02	1,96	10,35	373	145	766
PD11i	1,00	0,86	1,18	10	9	10	3,93	3,06	4,85	16	15	17	6,86	4,98	8,48	22	22	24
PD11ir	1,05	0,88	1,18	9	9	10	1,76	1,36	2,16	14	13	14	2,32	1,57	2,88	18	17	19
PD12i	1,75	1,53	2,00	10	9	10	5,56	4,22	6,72	16	15	16	9,36	6,97	11,49	22	22	23
PD21i	2,32	1,96	2,79	9	8	10	8,56	6,56	10,32	15	15	17	14,61	10,33	17,69	22	21	23
PD22i	4,44	3,80	5,11	8	8	9	12,37	9,88	14,79	15	14	16	19,92	15,04	24,07	21	21	22
PGls	3,51	1,12	6,03	421	132	722	10,73	3,04	18,58	1294	365	2231	19,16	5,12	34,66	2315	616	4163

Tabuľka B.13: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 12000$ a $n = 12000$.

Metóda	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	12,90	11,58	15,16	23	17	42	18,52	15,27	23,47	70	46	113	25,55	20,43	31,67	130	87	184
APG	1,31	0,92	1,70	135	95	174	8,81	5,88	11,56	907	605	1177	37,85	23,40	51,99	3893	2402	5299
APGls	2,65			156			17,84			1072			78,03			4653		
DPG1	2,16			129			14,06			849			60,29			3646		
DPG1ls	1,37	0,84	1,91	76	46	106	6,99	3,11	10,64	396	176	601	23,40	7,92	39,70	1333	451	2245
DPG2	2,22	1,57	2,87	135	95	174	14,85	9,91	19,39	907	605	1177	31,09	17,24	48,45	1899	1061	2937
DPG2ls	1,37	0,85	1,91	75	46	106	4,18	1,65	8,44	233	88	476	5,45	2,26	10,44	303	121	586
DPG3	2,22	1,58	2,86	135	95	174	14,82	9,92	19,42	907	605	1177	31,66	18,26	49,19	1938	1127	2985
DPG3ls	1,37	0,84	1,92	75	46	106	4,50	1,76	9,07	255	99	513	6,20	2,39	11,45	352	135	649
PD1i	1,21	1,02	1,37	9	9	10	4,60	3,52	5,41	16	15	17	8,03	5,86	10,12	22	21	23
PD1ir	1,29	1,09	1,45	9	9	10	2,21	1,74	2,61	14	13	14	2,96	1,95	3,73	18	17	19
PD12i	2,07	1,74	2,34	9	9	10	6,50	5,17	7,63	16	15	16	10,93	7,99	13,40	22	21	23
PD21i	2,74	2,34	3,16	9	8	10	9,90	7,43	11,81	15	14	16	17,14	12,44	22,10	22	21	23
PD22i	5,36	4,62	6,44	8	8	9	14,44	11,29	17,08	15	14	15	23,77	18,16	29,00	21	20	22
PGls	4,17	1,19	7,88	384	107	728	12,66	3,07	24,10	1173	283	2229	22,38	5,18	43,17	2078	479	4054

Tabuľka B.14: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 2400$ a $n = 3000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,29	0,26	0,33	27	21	34	0,59	0,49	0,71	83	65	106	0,97	0,81	1,15	155	124	193
	APG	0,03	0,02	0,03	75	56	96	0,15	0,11	0,20	449	326	596	0,58	0,40	0,82	1737	1209	2461
	APGls	0,04	0,03	0,05	72	49	96	0,24	0,14	0,32	436	271	600	0,90	0,50	1,33	1685	934	2495
	DPG1	0,04	0,03	0,05	75	56	96	0,24	0,18	0,32	449	326	596	0,94	0,64	1,36	1754	1212	2541
	DPG1ls	0,03	0,02	0,04	44	30	60	0,12	0,07	0,18	204	115	312	0,35	0,17	0,58	610	297	1019
	DPG2	0,04	0,03	0,05	75	56	96	0,23	0,14	0,32	425	260	596	0,32	0,19	0,49	593	346	917
	DPG2ls	0,03	0,02	0,04	44	30	60	0,06	0,04	0,08	95	59	136	0,09	0,05	0,12	141	86	208
	DPG3	0,04	0,03	0,05	75	56	96	0,23	0,15	0,32	430	279	596	0,34	0,20	0,52	631	374	964
	DPG3ls	0,03	0,02	0,04	44	30	60	0,06	0,04	0,09	107	66	150	0,10	0,06	0,14	175	101	249
	PD1i	0,05	0,04	0,06	9	9	10	0,18	0,15	0,22	16	15	16	0,28	0,22	0,34	22	21	22
	PD1ir	0,05	0,04	0,07	9	9	10	0,09	0,07	0,10	13	12	13	0,10	0,08	0,12	17	16	18
	PD12i	0,09	0,08	0,12	9	9	10	0,30	0,24	0,36	16	15	16	0,46	0,34	0,56	22	21	23
	PD13	1,78	1,68	1,94	9	9	10	3,00	2,82	3,21	16	15	17	4,20	3,96	4,42	22	21	23
	PD21i	0,11	0,09	0,12	8	8	9	0,37	0,30	0,44	15	14	15	0,58	0,44	0,72	21	21	21
	PD22i	0,24	0,20	0,29	8	8	9	0,64	0,52	0,78	14	14	15	0,99	0,75	1,25	21	20	21
	PD23	2,52	2,38	2,80	8	8	9	4,47	4,16	4,69	15	14	15	6,36	6,21	6,68	21	21	22
	PGls	0,05	0,03	0,09	138	65	255	0,16	0,07	0,32	432	185	881	0,28	0,12	0,58	775	305	1611

Tabuľka B.15: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 2400$ a $n = 6000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácii		čas		# iterácii		čas		# iterácii		čas		# iterácii		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,34	0,30	0,42	21	17	38	0,56	0,45	0,89	60	44	119	0,85	0,66	1,37	110	78	203
	APG	0,08	0,05	0,10	120	84	154	0,49	0,33	0,62	778	534	994	2,16	1,34	2,92	3458	2150	4677
	APGls	0,13	0,10	0,20	116	80	186	0,83	0,55	1,40	751	503	1274	3,83	2,21	7,34	3309	1965	6363
	DPG1	0,13	0,09	0,16	120	84	154	0,82	0,56	1,05	778	534	995	3,67	2,20	4,92	3495	2128	4762
	DPG1ls	0,07	0,05	0,11	59	37	85	0,32	0,17	0,48	285	149	418	1,09	0,45	1,77	937	401	1546
	DPG2	0,13	0,09	0,16	120	84	154	0,78	0,56	1,03	742	534	986	1,08	0,81	1,65	1022	765	1563
	DPG2ls	0,07	0,05	0,10	59	37	85	0,16	0,09	0,28	134	70	239	0,24	0,13	0,39	198	105	338
	DPG3	0,13	0,09	0,17	120	84	154	0,79	0,56	1,05	749	534	986	1,18	0,86	1,79	1119	821	1651
	DPG3ls	0,07	0,05	0,11	59	37	85	0,17	0,09	0,30	147	79	263	0,28	0,14	0,49	241	120	430
	PD1li	0,13	0,10	0,16	10	9	10	0,43	0,31	0,58	16	16	17	0,70	0,50	1,01	23	22	23
	PD1lir	0,13	0,11	0,17	10	9	10	0,18	0,13	0,23	14	13	14	0,21	0,15	0,27	18	17	18
	PD12i	0,25	0,19	0,30	10	9	11	0,72	0,53	0,92	16	16	17	1,11	0,80	1,57	23	22	23
	PD13	3,17	2,94	3,57	11	10	12	5,08	4,73	5,43	17	16	18	7,01	6,81	7,31	23	23	24
	PD21i	0,35	0,26	0,43	9	9	10	1,18	0,89	1,45	16	15	16	1,88	1,26	2,60	22	21	23
	PD22i	0,69	0,54	0,83	9	9	10	1,87	1,41	2,35	16	15	16	2,78	1,97	3,69	22	22	22
	PD23	5,06	4,67	5,50	9	9	10	8,40	7,81	8,91	16	15	17	11,71	11,28	12,32	22	22	23
	PGls	0,17	0,06	0,29	236	92	420	0,53	0,16	1,00	759	240	1479	0,96	0,27	1,86	1396	402	2754

Tabuľka B.16: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 2400$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácii		čas		čas		# iterácii		čas		čas		# iterácii		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,49	0,47	0,58	27	25	35	0,81	0,70	0,97	76	59	102	1,21	1,04	1,43	136	112	171
	APG	0,28	0,19	0,35	199	137	244	1,88	1,30	2,32	1336	923	1647	9,56	6,18	12,27	6815	4421	8703
	APGls	0,49	0,34	0,69	193	134	256	3,52	2,35	4,73	1301	874	1741	18,05	11,33	25,67	6600	4143	9345
	DPG1	0,49	0,35	0,61	199	137	244	3,24	2,23	4,01	1334	924	1643	16,65	10,68	21,30	6881	4459	8751
	DPG1ls	0,22	0,13	0,29	81	50	108	1,06	0,62	1,42	394	221	522	4,00	1,94	5,87	1417	676	2057
	DPG2	0,49	0,34	0,63	199	137	244	2,99	1,84	3,90	1229	758	1605	4,12	2,46	6,85	1691	1012	2824
	DPG2ls	0,22	0,13	0,28	81	50	108	0,51	0,30	0,95	193	111	364	0,76	0,43	1,27	285	159	471
	DPG3	0,49	0,33	0,63	199	137	244	3,01	1,89	3,93	1243	788	1605	4,58	3,02	6,87	1889	1244	2842
	DPG3ls	0,22	0,13	0,28	81	50	108	0,53	0,30	0,99	201	113	375	0,93	0,47	1,52	348	177	558
	PD1li	0,40	0,30	0,58	11	11	12	1,38	0,93	1,76	18	17	18	2,18	1,46	2,72	24	24	25
	PD1lir	0,41	0,32	0,60	11	11	12	0,49	0,36	0,65	15	14	16	0,54	0,40	0,70	19	18	20
	PD12i	0,74	0,57	0,96	11	11	12	2,06	1,46	2,56	18	17	18	3,22	2,20	3,94	24	24	25
	PD13	6,10	5,88	6,53	12	12	13	9,29	8,86	9,56	19	18	19	12,39	11,82	12,86	25	24	26
	PD2li	1,09	0,86	1,37	11	11	11	3,27	2,34	4,20	17	17	18	5,29	3,54	6,61	24	23	25
	PD22i	1,90	1,46	2,31	11	11	12	4,59	3,38	5,63	17	17	18	7,04	4,91	8,67	24	24	25
	PD23	10,31	9,98	11,01	11	11	12	16,14	15,31	16,54	18	17	18	21,75	20,73	22,70	24	23	25
	PGls	0,68	0,28	1,08	392	157	612	2,31	0,81	3,68	1339	475	2123	4,40	1,54	7,36	2555	888	4293

Tabuľka B.17: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 2400$ a $n = 24000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácií			čas			# iterácií			čas			# iterácií		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,93	0,74	1,03	34	30	38	1,45	1,17	1,61	94	81	109	2,05	1,68	2,31	166	140	191
	APG	1,03	0,73	1,30	316	226	400	7,02	5,11	8,87	2168	1584	2754	39,67	26,22	51,10	12285	8086	15871
	APGls	1,79			313			12,13			2148			68,95			12224		
	DPG1	1,82			345			12,65			2385			72,42			13661		
	DPG1ls	0,64	0,44	0,86	105	71	143	3,00	1,88	4,07	499	309	669	11,58	5,96	16,84	1924	986	2797
	DPG2	1,71	1,23	2,16	316	226	400	10,32	6,44	14,36	1921	1195	2687	14,11	9,41	24,26	2626	1749	4541
	DPG2ls	0,64	0,43	0,87	105	71	143	1,50	0,79	2,82	246	130	463	2,23	1,21	3,69	366	197	604
	DPG3	1,71	1,23	2,16	316	226	400	10,45	6,62	14,46	1945	1239	2705	16,05	11,26	24,69	2990	2103	4606
	DPG3ls	0,64	0,43	0,86	105	71	143	1,57	0,88	2,86	260	149	476	2,87	1,45	4,88	476	242	811
	PD1li	1,31	1,01	1,75	14	13	15	3,83	2,61	5,06	20	19	21	5,95	4,22	7,78	26	26	27
	PD1lr	1,26	0,84	1,77	13	12	15	1,36	0,94	1,85	16	16	17	1,43	0,99	1,93	21	20	22
	PD12i	2,02	1,60	2,57	13	13	14	5,11	3,67	6,66	20	19	21	7,80	5,49	10,00	26	25	27
	PD13	13,88	13,02	14,85	14	14	15	20,11	18,92	21,55	21	20	22	26,12	25,31	27,17	27	27	28
	PD2li	3,00	2,24	3,83	14	13	15	8,65	5,78	11,50	20	19	21	13,27	9,18	17,48	27	26	28
	PD22i	4,87	3,60	6,22	14	13	15	11,04	7,80	13,89	20	19	21	16,70	12,08	21,88	26	26	27
	PD23	25,60	23,35	27,76	14	13	15	36,97	35,47	38,82	20	20	21	48,50	46,54	51,19	27	26	28
	PGls	2,28	1,03	3,79	586	266	970	8,20	3,09	14,74	2108	796	3785	16,21	5,61	30,83	4167	1443	7922

Tabuľka B.20: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 1200$ a $n = 6000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,10	0,09	0,11	27	25	30	0,18	0,16	0,19	75	65	86	0,27	0,24	0,31	135	115	157
APG	0,05	0,04	0,07	145	101	181	0,34	0,23	0,44	922	629	1198	1,60	0,96	2,18	4396	2630	5956
APGls	0,09	0,06	0,12	138	93	201	0,53	0,34	0,81	868	564	1329	2,50	1,47	4,12	4123	2460	6809
DPG1	0,09	0,06	0,11	145	101	181	0,54	0,37	0,71	922	630	1197	2,63	1,57	3,58	4478	2689	6029
DPG1ls	0,04	0,02	0,05	60	38	78	0,18	0,10	0,25	273	158	388	0,60	0,28	0,91	937	447	1423
DPG2	0,09	0,06	0,11	145	101	181	0,46	0,24	0,59	783	411	1015	0,62	0,33	0,91	1046	558	1542
DPG2ls	0,04	0,02	0,05	60	38	78	0,09	0,05	0,17	138	76	255	0,14	0,07	0,24	204	111	357
DPG3	0,09	0,06	0,11	145	101	181	0,47	0,25	0,60	800	431	1015	0,70	0,47	0,98	1196	804	1661
DPG3ls	0,04	0,02	0,05	60	38	78	0,10	0,06	0,18	146	82	272	0,17	0,08	0,34	256	131	536
PD1i	0,11	0,08	0,14	11	10	11	0,33	0,23	0,41	17	17	18	0,52	0,36	0,67	24	23	24
PD1ir	0,11	0,09	0,16	11	10	11	0,13	0,10	0,17	14	14	15	0,15	0,12	0,19	19	18	19
PD12i	0,20	0,16	0,24	11	10	11	0,51	0,37	0,64	17	17	18	0,80	0,55	1,03	24	23	24
PD13	1,04	0,95	1,11	12	11	12	1,61	1,56	1,71	18	18	19	2,18	2,08	2,27	25	24	25
PD21i	0,25	0,17	0,32	11	10	11	0,72	0,54	0,92	17	16	17	1,11	0,79	1,47	23	23	24
PD22i	0,49	0,35	0,61	11	10	11	1,14	0,81	1,38	17	17	18	1,68	1,17	2,16	23	23	24
PD23	1,77	1,60	1,96	11	10	12	2,77	2,67	2,95	17	17	18	3,78	3,62	3,92	24	23	24
PGls	0,09	0,04	0,14	226	96	357	0,32	0,11	0,51	786	268	1274	0,61	0,20	1,03	1511	488	2542

Tabuľka B.21: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 2400$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácii		čas		čas		# iterácii		čas		čas		# iterácii		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,49	0,47	0,58	27	25	35	0,81	0,70	0,97	76	59	102	1,21	1,04	1,43	136	112	171
	APG	0,28	0,19	0,35	199	137	244	1,88	1,30	2,32	1336	923	1647	9,56	6,18	12,27	6815	4421	8703
	APGls	0,49	0,34	0,69	193	134	256	3,52	2,35	4,73	1301	874	1741	18,05	11,33	25,67	6600	4143	9345
	DPG1	0,49	0,35	0,61	199	137	244	3,24	2,23	4,01	1334	924	1643	16,65	10,68	21,30	6881	4459	8751
	DPG1ls	0,22	0,13	0,29	81	50	108	1,06	0,62	1,42	394	221	522	4,00	1,94	5,87	1417	676	2057
	DPG2	0,49	0,34	0,63	199	137	244	2,99	1,84	3,90	1229	758	1605	4,12	2,46	6,85	1691	1012	2824
	DPG2ls	0,22	0,13	0,28	81	50	108	0,51	0,30	0,95	193	111	364	0,76	0,43	1,27	285	159	471
	DPG3	0,49	0,33	0,63	199	137	244	3,01	1,89	3,93	1243	788	1605	4,58	3,02	6,87	1889	1244	2842
	DPG3ls	0,22	0,13	0,28	81	50	108	0,53	0,30	0,99	201	113	375	0,93	0,47	1,52	348	177	558
	PD1li	0,40	0,30	0,58	11	11	12	1,38	0,93	1,76	18	17	18	2,18	1,46	2,72	24	24	25
	PD1lir	0,41	0,32	0,60	11	11	12	0,49	0,36	0,65	15	14	16	0,54	0,40	0,70	19	18	20
	PD12i	0,74	0,57	0,96	11	11	12	2,06	1,46	2,56	18	17	18	3,22	2,20	3,94	24	24	25
	PD13	6,10	5,88	6,53	12	12	13	9,29	8,86	9,56	19	18	19	12,39	11,82	12,86	25	24	26
	PD2li	1,09	0,86	1,37	11	11	11	3,27	2,34	4,20	17	17	18	5,29	3,54	6,61	24	23	25
	PD22i	1,90	1,46	2,31	11	11	12	4,59	3,38	5,63	17	17	18	7,04	4,91	8,67	24	24	25
	PD23	10,31	9,98	11,01	11	11	12	16,14	15,31	16,54	18	17	18	21,75	20,73	22,70	24	23	25
	PGls	0,68	0,28	1,08	392	157	612	2,31	0,81	3,68	1339	475	2123	4,40	1,54	7,36	2555	888	4293

Tabuľka B.22: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 4800$ a $n = 24000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácii			čas			# iterácii			čas			# iterácii		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	2,97	2,50	3,24	17	14	22	4,04	3,22	5,35	59	34	108	5,55	3,80	8,40	119	57	230
	APG	1,86	1,12	2,84	265	163	339	13,07	8,20	19,00	1874	1202	2357	70,04	39,04	105,08	10092	5717	13720
	APGls	2,48			212			17,44			1507			87,74			7600		
	DPG1	2,62			234			18,86			1680			97,46			8694		
	DPG1ls	1,35	0,77	1,98	108	61	156	6,76	3,04	10,35	548	248	796	25,36	8,52	41,36	2063	697	3386
	DPG2	3,03	1,85	4,23	265	163	339	21,02	12,98	28,72	1838	1156	2357	31,45	15,63	50,59	2748	1390	4481
	DPG2ls	1,35	0,77	1,94	108	61	156	3,48	1,57	5,93	278	124	483	4,94	2,22	7,80	394	175	632
	DPG3	3,01	1,83	3,91	265	163	339	20,96	13,09	28,16	1844	1169	2357	33,66	16,77	52,30	2959	1495	4542
	DPG3ls	1,34	0,77	1,94	108	61	156	3,67	1,65	6,22	299	133	508	5,88	2,29	12,42	480	187	1013
	PD1i	1,78	1,32	2,24	12	11	12	5,91	3,62	8,28	18	18	19	9,79	5,85	13,49	25	24	27
	PD1ir	1,60	1,27	2,28	11	11	12	2,06	1,52	2,60	15	15	16	2,33	1,63	2,98	20	19	21
	PD12i	2,82	1,91	3,57	12	11	12	7,82	5,31	10,78	18	17	19	12,54	7,68	17,02	24	24	25
	PD13	43,49			13			66,78			20			87,50			26		
	PD21i	4,06	2,63	5,49	11	11	12	13,10	8,23	17,45	18	17	18	21,49	12,48	30,60	24	24	28
	PD22i	7,15	4,96	9,17	12	11	13	17,52	10,93	22,11	18	17	19	27,20	16,95	37,21	25	24	25
	PD23	78,48			12			117,57			18			161,69			25		
	PGls	5,18	1,37	9,20	659	174	1173	17,17	3,74	34,84	2190	478	4453	32,62	6,45	69,85	4164	825	8930

Tabuľka B.23: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 9600$ a $n = 48000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácií			čas			# iterácií			čas			# iterácií		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	21,99	21,35	25,39	27	23	59	27,21	25,25	34,86	78	61	149	33,70	30,32	44,17	140	109	239
	APG	13,12			440			95,78		3205				562,79		18818			
	DPG1ls	7,49	3,84	11,29	139	71	208	40,09	17,38	63,26	754	326	1173	157,43	53,37	275,15	2966	1007	5162
	DPG2	21,84			440			158,34		3205				385,97		7822			
	DPG2ls	7,46	3,86	11,19	139	71	208	22,46	6,25	47,29	420	113	885	30,13	8,92	60,94	562	161	1138
	DPG3	21,70			440			158,00		3205				387,42		7853			
	DPG3ls	7,45	3,84	11,13	139	71	208	23,95	7,64	49,57	451	143	930	36,07	12,01	74,61	680	226	1401
	PD1i	6,90	5,70	8,86	13	12	16	23,52	15,04	32,51	20	18	23	39,94	23,12	56,16	26	25	29
	PD1lr	6,38	5,04	8,22	13	11	16	8,82	5,95	11,22	17	15	20	10,35	6,38	13,40	21	20	25
	PD12i	10,76	8,32	13,79	12	12	13	30,48	19,83	40,89	19	18	19	50,32	32,66	68,34	25	24	26
	PD21i	19,04			12			61,28		18				112,06		25			
	PD22i	33,37			14			80,59		20				138,96		27			
	PGls	58,60			1788			203,03		6189				395,60		12063			

Tabuľka B.24: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 0.1% ($de = 0.001$).

Metóda	Presn. ϵ			10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,27	0,20	0,86	43	26	177	0,68	0,40	2,50	135	71	550	1,92	0,69	14,33	413	138	3179			
APG	0,01	0,01	0,01	36	26	44	0,05	0,03	0,09	185	116	353	0,19	0,11	0,29	783	475	1258			
APGls	0,01	0,01	0,02	31	21	39	0,06	0,03	0,11	157	91	291	0,24	0,13	0,39	650	361	1073			
DPG1	0,01	0,01	0,02	36	26	44	0,07	0,04	0,12	186	116	353	0,28	0,17	0,46	815	484	1362			
DPG1ls	0,01	0,01	0,01	22	15	26	0,04	0,03	0,08	100	60	202	0,15	0,09	0,28	382	230	684			
DPG2	0,01	0,01	0,02	36	26	44	0,06	0,04	0,13	181	110	384	0,15	0,07	0,45	413	197	1313			
DPG2ls	0,01	0,01	0,01	22	15	26	0,04	0,02	0,09	97	52	230	0,09	0,03	0,29	215	87	706			
DPG3	0,01	0,01	0,02	36	26	44	0,07	0,04	0,18	189	116	499	0,16	0,08	0,48	448	217	1385			
DPG3ls	0,01	0,01	0,01	22	15	26	0,04	0,02	0,11	108	55	283	0,10	0,04	0,32	245	96	761			
PD11i	0,19	0,11	0,39	16	14	19	1,54	0,90	2,73	23	20	27	2,31	1,27	4,46	29	26	33			
PD11ir	0,14	0,08	0,23	16	13	19	0,31	0,16	0,46	20	18	24	0,35	0,18	0,54	25	22	28			
PD12i	0,37	0,24	0,51	18	15	21	2,94	1,59	5,73	26	22	30	4,31	2,27	8,51	33	28	36			
PD13	2,72	2,32	3,44	13	12	14	4,11	3,60	5,02	19	18	20	6,01	4,96	7,38	26	25	26			
PD21i	0,39	0,22	0,78	16	14	18	2,73	1,51	4,68	22	20	24	4,05	2,12	7,59	29	26	31			
PD22i	0,97	0,67	1,34	19	16	24	4,66	2,45	8,39	27	22	31	6,52	3,50	12,58	33	29	38			
PD23	2,44	2,10	3,18	11	11	12	3,86	3,23	4,78	18	17	19	5,78	5,02	7,31	24	24	25			
PGls	0,01	0,01	0,02	46	26	61	0,17	0,04	0,78	591	156	2756	1,04	0,12	10,07	3748	434	36182			

Tabuľka B.25: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 0.5% ($de = 0.005$).

Metóda	Presn. ϵ			10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,28	0,27	0,29	24	22	26	0,50	0,47	0,55	67	61	76	0,77	0,71	0,85	119	107	136			
APG	0,02	0,02	0,03	72	61	88	0,12	0,10	0,15	410	346	514	0,55	0,43	0,74	1831	1492	2538			
APGls	0,03	0,02	0,05	58	37	94	0,16	0,09	0,26	320	181	533	0,68	0,35	1,25	1387	723	2595			
DPG1	0,03	0,03	0,04	72	61	88	0,19	0,16	0,23	410	346	514	0,86	0,69	1,17	1845	1496	2566			
DPG1ls	0,02	0,01	0,02	33	27	42	0,07	0,06	0,11	144	110	199	0,25	0,19	0,39	485	359	736			
DPG2	0,03	0,03	0,04	72	61	88	0,19	0,16	0,23	406	340	514	0,32	0,21	0,50	687	447	1053			
DPG2ls	0,02	0,01	0,02	33	27	42	0,05	0,03	0,06	86	63	116	0,07	0,05	0,10	135	99	187			
DPG3	0,03	0,03	0,04	72	61	88	0,19	0,16	0,23	407	346	514	0,34	0,23	0,51	736	500	1081			
DPG3ls	0,02	0,01	0,02	33	27	42	0,05	0,03	0,06	88	64	118	0,08	0,06	0,14	162	110	248			
PD11i	0,14	0,11	0,19	11	11	12	0,72	0,59	0,88	18	17	18	1,11	0,93	1,44	24	23	25			
PD11lr	0,12	0,09	0,14	11	10	12	0,19	0,15	0,24	15	14	16	0,23	0,19	0,29	19	19	20			
PD12i	0,29	0,22	0,38	12	11	12	1,24	1,04	1,44	18	17	19	1,86	1,49	2,48	24	24	25			
PD13	5,89	4,76	6,65	13	12	13	8,94	7,57	9,98	19	18	20	12,06	9,98	14,34	25	25	26			
PD21i	0,36	0,24	0,45	11	10	12	1,62	1,34	2,08	17	17	18	2,50	2,03	3,34	24	23	24			
PD22i	0,76	0,65	0,96	11	11	12	2,35	1,93	2,93	17	17	19	3,43	2,70	4,67	24	23	25			
PD23	5,60	4,53	6,27	12	11	13	8,66	7,00	9,40	18	17	19	11,91	9,89	14,06	24	24	25			
PGLs	0,03	0,02	0,04	84	62	124	0,11	0,08	0,21	347	222	657	0,23	0,15	0,44	703	449	1355			

Tabuľka B.26: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 1% ($de = 0.01$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,48	0,42	0,55	25	23	27	0,72	0,65	0,81	69	62	76	1,01	0,91	1,12	123	108	135
APG	0,04	0,03	0,04	98	80	117	0,22	0,18	0,27	581	469	693	1,04	0,78	1,26	2695	2035	3288
APGls	0,06	0,04	0,09	87	61	128	0,34	0,24	0,54	507	350	770	1,55	0,98	2,48	2321	1439	3841
DPG1	0,06	0,05	0,07	98	80	117	0,37	0,30	0,44	581	469	693	1,73	1,35	2,09	2720	2125	3311
DPG1ls	0,03	0,03	0,04	42	32	51	0,14	0,10	0,17	189	129	233	0,45	0,31	0,57	635	440	823
DPG2	0,06	0,05	0,07	98	80	117	0,36	0,30	0,43	574	469	693	0,59	0,42	0,93	922	655	1472
DPG2ls	0,03	0,02	0,04	42	32	51	0,08	0,06	0,10	106	79	144	0,12	0,09	0,14	161	122	202
DPG3	0,06	0,05	0,07	98	80	117	0,36	0,30	0,44	576	469	693	0,63	0,48	0,95	992	769	1494
DPG3ls	0,03	0,02	0,04	42	32	51	0,08	0,06	0,11	111	85	154	0,14	0,09	0,21	200	133	290
PD11i	0,16	0,13	0,20	11	11	12	0,72	0,54	0,85	18	17	18	1,11	0,86	1,32	24	24	25
PD11ir	0,15	0,13	0,19	11	11	12	0,21	0,18	0,24	15	14	16	0,24	0,21	0,28	19	18	20
PD12i	0,31	0,25	0,40	11	11	12	1,13	0,91	1,29	18	17	18	1,70	1,37	1,99	24	24	25
PD13	3,65	3,52	3,86	12	12	13	5,59	5,30	5,87	19	18	20	7,45	7,36	7,73	25	25	26
PD21i	0,43	0,30	0,48	11	10	11	1,64	1,35	1,99	17	17	18	2,52	1,97	3,00	24	23	24
PD22i	0,83	0,70	1,02	11	10	12	2,28	1,86	2,68	17	17	18	3,37	2,73	3,89	24	23	24
PD23	3,70	3,56	3,95	11	11	12	5,82	5,55	5,94	18	17	18	7,90	7,54	8,21	24	23	25
PGls	0,06	0,04	0,08	126	83	181	0,21	0,14	0,31	488	303	698	0,42	0,27	0,63	959	609	1445

Tabuľka B.27: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 5% ($de = 0.05$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií			
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,49	0,47	0,58	27	25	35	0,81	0,70	0,97	76	59	102	1,21	1,04	1,43	136	112	171
APG	0,28	0,19	0,35	199	137	244	1,88	1,30	2,32	1336	923	1647	9,56	6,18	12,27	6815	4421	8703
APGls	0,49	0,34	0,69	193	134	256	3,52	2,35	4,73	1301	874	1741	18,05	11,33	25,67	6600	4143	9345
DPG1	0,49	0,35	0,61	199	137	244	3,24	2,23	4,01	1334	924	1643	16,65	10,68	21,30	6881	4459	8751
DPG1ls	0,22	0,13	0,29	81	50	108	1,06	0,62	1,42	394	221	522	4,00	1,94	5,87	1417	676	2057
DPG2	0,49	0,34	0,63	199	137	244	2,99	1,84	3,90	1229	758	1605	4,12	2,46	6,85	1691	1012	2824
DPG2ls	0,22	0,13	0,28	81	50	108	0,51	0,30	0,95	193	111	364	0,76	0,43	1,27	285	159	471
DPG3	0,49	0,33	0,63	199	137	244	3,01	1,89	3,93	1243	788	1605	4,58	3,02	6,87	1889	1244	2842
DPG3ls	0,22	0,13	0,28	81	50	108	0,53	0,30	0,99	201	113	375	0,93	0,47	1,52	348	177	558
PD1li	0,40	0,30	0,58	11	11	12	1,38	0,93	1,76	18	17	18	2,18	1,46	2,72	24	24	25
PD1lir	0,41	0,32	0,60	11	11	12	0,49	0,36	0,65	15	14	16	0,54	0,40	0,70	19	18	20
PD12i	0,74	0,57	0,96	11	11	12	2,06	1,46	2,56	18	17	18	3,22	2,20	3,94	24	24	25
PD13	6,10	5,88	6,53	12	12	13	9,29	8,86	9,56	19	18	19	12,39	11,82	12,86	25	24	26
PD2li	1,09	0,86	1,37	11	11	11	3,27	2,34	4,20	17	17	18	5,29	3,54	6,61	24	23	25
PD22i	1,90	1,46	2,31	11	11	12	4,59	3,38	5,63	17	17	18	7,04	4,91	8,67	24	24	25
PD23	10,31	9,98	11,01	11	11	12	16,14	15,31	16,54	18	17	18	21,75	20,73	22,70	24	23	25
PGls	0,68	0,28	1,08	392	157	612	2,31	0,81	3,68	1339	475	2123	4,40	1,54	7,36	2555	888	4293

Tabuľka B.28: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 10% ($de = 0.1$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Presn. ϵ																		
ADMM	0,51	0,43	0,59	17	13	26	0,82	0,62	1,26	55	32	106	1,25	0,82	2,27	108	54	231
APG	0,77	0,45	1,05	252	150	351	5,40	3,23	7,24	1792	1082	2428	28,38	14,82	41,44	9446	4980	13918
APGls	1,18			222			7,91			1493			40,59			7668		
DPG1	1,44			292			10,17			2068			55,41			11260		
DPG1ls	0,56	0,27	0,90	101	49	160	2,82	1,14	4,43	505	209	794	10,31	2,96	18,82	1848	535	3395
DPG2	1,26	0,75	1,75	252	150	351	8,13	4,66	11,96	1632	930	2428	10,75	6,42	18,30	2154	1275	3717
DPG2ls	0,56	0,28	0,87	101	49	160	1,57	0,50	3,02	282	89	547	2,17	0,69	3,82	386	122	689
DPG3	1,27	0,75	1,75	252	150	351	8,23	4,88	11,98	1652	975	2428	11,97	6,83	19,66	2405	1356	3998
DPG3ls	0,56	0,27	0,87	101	49	160	1,64	0,53	3,13	296	98	566	2,51	0,80	5,49	451	147	989
PD1li	0,65	0,46	0,89	11	11	12	2,02	1,15	2,84	18	17	18	3,24	1,76	4,87	24	24	25
PD1liir	0,63	0,49	0,88	11	10	12	0,76	0,53	0,97	15	14	16	0,85	0,55	1,07	19	18	20
PD12i	1,12	0,77	1,48	11	11	12	2,99	1,80	3,96	18	17	18	4,64	2,63	6,67	24	24	25
PD13	6,55	6,13	7,13	12	12	13	9,96	9,25	10,63	19	18	20	13,34	12,70	14,05	25	24	26
PD2li	1,58	1,01	2,16	11	10	11	4,55	2,66	6,57	17	17	18	7,27	3,78	11,06	24	23	24
PD22i	2,74	1,93	3,65	11	11	12	6,51	3,85	8,77	18	17	18	10,04	5,70	14,29	24	24	24
PD23	11,35	10,49	12,32	11	11	12	17,69	16,70	18,49	18	17	19	23,84	22,63	24,92	24	24	25
PGLs	2,14	0,52	4,48	589	144	1237	6,88	1,29	16,05	1897	357	4451	12,82	2,12	32,57	3536	585	9024

Tabuľka B.29: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 20% ($de = 0.2$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,63	0,50	1,94	22	13	140	1,06	0,73	3,43	61	33	276	1,59	1,01	4,91	110	57	411
APG	1,86	0,90	2,97	310	150	497	13,83	6,76	21,32	2318	1137	3590	75,79	34,44	127,23	12725	5814	21448
DPG1ls	1,27	0,48	2,40	119	45	225	6,56	1,74	13,18	620	165	1245	24,34	4,04	57,80	2302	385	5502
DPG2	3,01	1,47	4,81	310	150	497	18,20	4,65	29,86	1881	484	3103	23,77	7,43	47,55	2455	772	4901
DPG2ls	1,27	0,48	2,39	119	45	225	3,66	0,80	8,41	343	74	797	4,84	1,02	10,57	452	93	996
DPG3	3,02	1,46	4,83	310	150	497	17,99	5,57	29,96	1860	574	3103	25,85	9,04	44,58	2673	932	4622
DPG3ls	1,27	0,48	2,39	119	45	225	3,86	0,81	8,76	364	77	828	5,74	1,19	14,36	542	113	1359
PD1li	1,01	0,59	1,47	11	10	12	2,80	1,57	4,68	18	17	18	4,51	2,29	7,80	24	23	25
PD1lir	1,02	0,65	1,48	11	10	11	1,23	0,68	2,03	15	13	16	1,34	0,70	2,24	19	18	20
PD12i	1,77	1,04	2,55	11	10	12	4,25	2,46	6,83	18	17	18	6,75	3,54	11,17	24	23	25
PD13	6,63	5,22	7,13	12	10	13	10,03	8,28	10,48	19	16	20	13,36	11,88	13,96	25	23	26
PD2li	2,32	1,39	3,36	11	10	12	6,10	3,42	10,46	17	17	18	9,89	5,17	17,07	24	23	26
PD22i	4,10	2,32	7,37	12	11	28	9,31	5,22	14,54	18	17	34	14,43	7,59	23,65	24	24	41
PD23	11,21	9,97	12,31	12	10	12	17,24	15,69	18,42	18	16	19	23,42	22,32	24,65	24	23	25
PGls	5,72	0,82	16,14	834	119	2369	17,69	1,56	54,80	2588	228	8048	33,07	2,35	112,32	4840	345	16500

Tabuľka B.30: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 40% ($de = 0.4$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	2,91	0,62	8,56	138	14	443	5,16	0,94	12,89	260	31	673	7,47	1,53	18,17	386	63	958
APG	4,51	2,73	11,16	339	207	841	34,54	19,66	80,79	2603	1462	6120	204,74	117,84	521,80	15438	8808	39540
DPG1ls	2,54	1,16	9,01	109	50	389	12,80	3,90	51,86	557	167	2262	47,00	8,56	249,38	2049	368	10873
DPG2	7,24	4,36	17,91	339	207	841	22,55	10,54	94,49	1055	486	4421	33,78	16,61	117,82	1579	767	5508
DPG2ls	2,53	1,16	8,99	109	50	389	5,97	1,78	28,40	257	75	1231	7,98	2,32	37,13	342	96	1609
DPG3	7,25	4,39	18,11	339	207	841	23,26	11,71	93,41	1089	544	4409	42,91	22,31	160,30	2010	1034	7501
DPG3ls	2,54	1,17	8,97	109	50	389	6,44	1,94	29,98	279	83	1307	9,48	2,67	42,08	412	114	1829
PD11i	1,56	1,05	3,09	12	10	15	3,69	2,43	8,20	18	17	21	5,84	3,44	14,30	25	23	28
PD11ir	1,73	1,21	3,12	12	10	15	1,97	1,24	4,09	15	13	18	2,08	1,25	4,48	19	17	22
PD12i	2,74	1,89	4,80	12	10	15	5,92	3,82	12,92	18	16	22	9,28	5,82	23,18	25	23	29
PD13	6,44	5,18	7,54	12	10	14	9,86	8,40	11,31	18	16	21	13,36	11,80	17,03	25	22	30
PD21i	3,44	2,52	5,72	12	10	16	7,93	5,11	17,71	18	16	22	12,38	7,58	30,28	25	23	30
PD22i	8,68	4,65	64,96	14	10	113	16,25	9,56	71,24	20	17	119	23,82	13,09	77,48	27	23	125
PD23	11,53	9,78	14,81	12	10	15	17,64	15,61	21,47	18	16	22	23,84	21,82	26,98	25	23	28
PGIs	13,26	2,08	94,00	898	139	6375	39,06	3,69	324,99	2651	245	22058	70,94	5,12	633,97	4810	342	42721

Tabuľka B.31: Vplyv riedkosti matice dát 1: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 100% ($de = 1$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	33,95	0,65	48,21	1862	15	2649	41,38	1,17	53,18	2274	43	2936	48,79	1,75	66,56	2684	75	3677
APG	10,94			838			86,07			6598			616,43			47289		
DPG1s	3,26	1,43	20,62	156	68	1002	15,32	3,32	131,28	745	160	6395	59,83	7,72	706,40	2914	374	34428
DPG2	11,30	7,14	41,65	576	365	2110	29,08	19,83	108,14	1481	1012	5476	46,48	29,44	167,51	2366	1501	8484
DPG2ls	3,28	1,44	20,82	157	68	1001	5,32	1,94	39,77	254	91	1917	7,34	2,29	59,03	350	107	2848
DPG3	11,34	7,14	41,75	578	365	2116	33,81	24,04	116,97	1723	1228	5931	60,29	39,76	187,86	3074	2031	9524
DPG3ls	3,37	1,42	22,31	162	68	1079	6,20	2,01	47,65	300	96	2307	9,05	2,58	67,37	438	124	3263
PD11i	2,72	1,04	3,85	24	11	32	4,19	1,84	7,79	30	17	38	5,62	2,84	12,95	37	24	45
PD11ir	3,08	1,19	4,38	23	11	31	3,33	1,23	4,78	27	14	34	3,36	1,23	5,19	31	18	38
PD12i	4,64	1,88	10,30	24	11	48	7,01	3,17	12,46	31	17	54	9,09	4,61	20,13	37	24	60
PD13	12,30	4,77	16,50	23	9	31	15,66	8,00	19,84	29	15	37	19,20	11,47	23,62	36	22	44
PD21i	6,95	2,21	10,48	24	10	33	10,47	4,28	15,92	31	17	40	13,81	6,40	28,96	38	23	47
PD22i	13,44	4,78	54,74	27	10	97	19,40	7,97	59,88	33	16	104	24,75	11,00	63,38	40	23	110
PD23	22,71	9,70	30,49	23	10	31	28,96	15,44	36,41	30	16	37	35,60	22,27	43,09	36	23	44
PGls	29,09			2078			88,39			6326			161,10			11535		

Tabuľka B.32: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 0.1% ($de = 0.001$).

Metóda	Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,66	0,55	0,88	131	107	183	2,12	1,35	5,40	460	285	1207	3,48	2,12	13,12	767	464	2942	
APG	0,06	0,05	0,08	251	208	339	0,29	0,24	0,42	1192	979	1694	1,18	0,91	1,74	4836	3669	7120	
APGls	0,09	0,07	0,14	245	199	390	0,44	0,35	0,78	1160	911	1966	1,78	1,32	3,32	4718	3374	8776	
DPG1	0,09	0,07	0,11	251	208	339	0,41	0,34	0,58	1186	978	1701	1,70	1,32	2,50	4922	3757	7361	
DPG1ls	0,08	0,06	0,10	186	150	268	0,29	0,22	0,42	719	550	1006	0,60	0,37	0,91	1510	966	2308	
DPG2	0,09	0,07	0,12	251	208	339	0,29	0,20	0,40	824	558	1167	0,38	0,27	0,59	1091	777	1687	
DPG2ls	0,08	0,06	0,10	185	150	268	0,16	0,14	0,21	391	328	529	0,20	0,17	0,26	484	387	634	
DPG3	0,09	0,07	0,11	251	208	339	0,29	0,20	0,41	831	562	1176	0,44	0,32	0,61	1266	925	1770	
DPG3ls	0,08	0,06	0,11	186	150	268	0,17	0,13	0,23	410	322	539	0,21	0,16	0,27	514	396	650	
PD11i	0,15	0,14	0,18	14	13	16	0,24	0,21	0,30	21	20	22	0,34	0,30	0,44	27	26	28	
PD11ir	0,16	0,14	0,18	14	13	16	0,17	0,14	0,19	18	17	20	0,17	0,15	0,20	23	22	24	
PD12i	0,43	0,37	0,49	16	15	17	0,67	0,58	0,84	23	21	24	0,98	0,86	1,24	29	28	30	
PD13	3,08	2,71	3,86	14	14	15	4,71	4,17	6,00	21	20	22	7,25	6,29	8,94	27	27	28	
PD21i	0,28	0,24	0,33	14	13	15	0,44	0,40	0,55	20	20	22	0,63	0,56	0,80	27	26	28	
PD22i	0,59	0,51	0,68	16	15	17	0,92	0,79	1,15	22	21	24	1,34	1,16	1,69	29	28	30	
PD23	3,01	2,60	3,91	14	13	15	4,62	4,06	5,77	21	20	22	7,18	6,18	8,71	27	26	28	
PGls	0,58	0,40	1,10	2022	1350	3918	1,50	0,87	2,84	5234	3061	9983	1,81	1,14	3,45	6324	3936	12049	

Tabuľka B.33: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 0.5% ($de = 0.005$).

Presn. ϵ	Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
		čas		# iterácii		čas		čas		# iterácii		čas		čas		# iterácii		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,29	0,28	0,31	25	24	30	0,62	0,52	0,96	90	69	153	1,60	0,86	3,08	279	135	565
	APG	0,04	0,03	0,06	120	85	184	0,19	0,12	0,32	623	398	1035	0,70	0,40	1,23	2294	1360	4006
	APGls	0,06	0,04	0,11	119	83	198	0,32	0,19	0,60	620	391	1145	1,16	0,65	2,24	2279	1288	4456
	DPG1	0,06	0,04	0,09	120	85	184	0,29	0,19	0,49	623	403	1030	1,09	0,64	1,92	2320	1372	4069
	DPG1ls	0,05	0,03	0,08	86	49	147	0,16	0,10	0,38	287	180	682	0,31	0,22	0,69	562	408	1266
	DPG2	0,06	0,04	0,09	120	85	184	0,26	0,12	0,48	543	262	1007	0,32	0,17	0,56	680	347	1170
	DPG2ls	0,05	0,03	0,08	85	49	147	0,09	0,05	0,15	157	95	273	0,11	0,07	0,20	191	122	349
	DPG3	0,06	0,04	0,09	120	85	184	0,26	0,13	0,48	554	274	1011	0,34	0,18	0,59	729	367	1235
	DPG3ls	0,05	0,03	0,08	85	49	147	0,09	0,06	0,17	164	103	280	0,11	0,07	0,19	206	134	329
	PD11i	0,08	0,07	0,09	14	13	15	0,14	0,12	0,15	20	20	21	0,20	0,18	0,22	27	26	27
	PD11ir	0,09	0,08	0,10	14	13	15	0,09	0,09	0,11	18	17	18	0,10	0,09	0,12	22	21	23
	PD12i	0,20	0,17	0,24	14	13	15	0,35	0,32	0,39	20	20	21	0,54	0,51	0,60	27	26	27
	PD13	6,67	5,79	7,37	14	13	15	9,77	8,30	10,63	21	20	21	13,63	11,25	15,16	27	26	28
	PD21i	0,19	0,17	0,22	13	13	15	0,32	0,28	0,35	20	19	21	0,48	0,45	0,53	26	26	27
	PD22i	0,34	0,30	0,43	13	13	15	0,60	0,55	0,68	20	19	21	0,94	0,89	1,05	26	25	28
	PD23	6,66	5,49	7,34	14	13	15	9,77	8,46	10,73	20	19	21	13,69	11,60	15,16	27	26	28
	PGls	0,18	0,06	0,51	514	185	1484	0,37	0,12	1,16	1060	352	3317	0,44	0,18	1,22	1253	488	3487

Tabuľka B.34: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 1% ($de = 0.01$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,51	0,45	0,58	31	29	34	0,73	0,64	0,81	70	66	77	0,96	0,86	1,05	113	105	122
APG	0,03	0,03	0,04	83	78	87	0,18	0,16	0,19	437	409	458	0,65	0,61	0,70	1608	1533	1708
APGls	0,08	0,06	0,09	107	83	123	0,41	0,29	0,49	587	433	687	1,59	1,14	1,92	2289	1630	2748
DPG1	0,05	0,05	0,06	83	78	87	0,29	0,26	0,30	437	409	455	1,07	1,00	1,14	1626	1541	1724
DPG1ls	0,04	0,03	0,05	48	45	52	0,13	0,12	0,15	173	163	181	0,34	0,30	0,36	453	417	480
DPG2	0,05	0,05	0,06	83	78	87	0,27	0,19	0,30	414	293	455	0,39	0,26	0,61	590	396	926
DPG2ls	0,04	0,03	0,04	48	45	52	0,08	0,06	0,09	97	85	119	0,10	0,08	0,13	131	117	156
DPG3	0,05	0,05	0,06	83	78	87	0,27	0,20	0,30	418	308	455	0,41	0,29	0,62	625	445	948
DPG3ls	0,04	0,03	0,04	48	45	52	0,08	0,07	0,10	100	90	126	0,11	0,10	0,13	147	133	163
PD11i	0,07	0,06	0,08	13	12	14	0,14	0,13	0,15	19	18	21	0,23	0,22	0,26	25	25	27
PD11ir	0,08	0,07	0,09	13	12	14	0,09	0,08	0,10	16	15	18	0,10	0,09	0,11	21	20	22
PD12i	0,16	0,14	0,19	12	12	13	0,34	0,30	0,37	19	18	20	0,58	0,54	0,62	25	25	26
PD13	3,77	3,53	3,88	13	12	13	5,69	5,60	5,97	19	19	20	7,57	7,38	7,76	26	25	26
PD21i	0,18	0,16	0,21	13	12	17	0,35	0,32	0,38	20	18	23	0,56	0,52	0,61	26	25	30
PD22i	0,30	0,27	0,34	12	12	13	0,64	0,59	0,69	19	18	20	1,09	1,03	1,17	25	25	26
PD23	4,01	3,89	4,28	12	12	13	6,14	5,84	6,26	19	18	19	8,22	7,90	8,54	25	24	26
PGls	0,08	0,06	0,09	160	143	177	0,17	0,15	0,19	360	323	390	0,27	0,24	0,30	567	530	611

Tabuľka B.35: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 5% ($de = 0.05$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,64	0,58	0,71	44	42	46	0,80	0,74	0,88	68	64	71	0,96	0,89	1,04	93	89	97
APG	0,14	0,13	0,15	98	94	101	0,85	0,82	0,89	608	583	630	4,07	3,88	4,32	2907	2762	3068
APGls	0,35	0,32	0,38	132	126	138	2,34	2,21	2,48	865	821	921	11,93	11,14	13,09	4376	4087	4816
DPG1	0,24	0,23	0,25	98	94	101	1,48	1,43	1,53	608	583	630	7,04	6,63	7,49	2917	2746	3095
DPG1ls	0,17	0,16	0,18	57	54	58	0,73	0,65	0,79	263	228	276	2,63	2,49	2,79	923	871	992
DPG2	0,24	0,23	0,25	98	94	101	1,46	1,17	1,53	599	477	630	2,52	1,66	3,82	1040	672	1572
DPG2ls	0,17	0,15	0,18	57	54	58	0,39	0,34	0,52	137	119	188	0,58	0,52	0,69	207	186	253
DPG3	0,24	0,23	0,25	98	94	101	1,46	1,22	1,54	601	502	630	2,67	1,87	3,89	1105	769	1608
DPG3ls	0,17	0,15	0,18	57	54	58	0,42	0,37	0,54	148	133	204	0,70	0,58	1,01	252	212	375
PD11i	0,23	0,19	0,27	12	10	15	0,64	0,57	0,70	18	17	21	1,12	1,03	1,24	25	23	28
PD11ir	0,25	0,21	0,29	12	10	15	0,30	0,26	0,34	15	14	18	0,33	0,29	0,37	20	18	23
PD12i	0,39	0,34	0,45	10	10	11	1,06	0,96	1,14	17	16	18	1,79	1,64	1,92	23	23	24
PD13	5,22	4,96	5,82	10	10	11	8,50	7,99	8,96	17	16	18	11,65	11,42	12,34	23	23	24
PD21i	0,57	0,49	0,64	11	10	12	1,59	1,49	1,72	18	17	19	2,76	2,53	2,93	24	23	26
PD22i	0,88	0,79	0,97	11	10	12	2,29	2,06	2,45	18	16	19	3,89	3,57	4,31	24	23	25
PD23	10,18	9,34	10,80	11	10	11	15,99	14,99	17,35	17	16	18	22,00	21,15	23,13	24	23	24
PGLs	0,40	0,34	0,44	225	204	245	1,41	1,29	1,52	803	746	875	2,67	2,48	2,91	1530	1424	1630

Tabuľka B.36: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 10% ($de = 0.1$).

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,79	0,71	0,85	52	50	54	0,93	0,83	1,01	69	64	73	1,08	0,96	1,17	87	80	92	1,08	0,96	1,17	87	80	92
APG	0,33	0,31	0,34	107	103	111	2,15	2,06	2,26	707	676	741	11,43	10,77	11,98	3783	3579	3976	11,43	10,77	11,98	3783	3579	3976
APGls	0,77	0,73	0,80	142	137	147	5,21	4,96	5,52	981	933	1039	28,98	28,00	30,54	5474	5284	5771	28,98	28,00	30,54	5474	5284	5771
DPG1	0,54	0,52	0,56	107	103	111	3,54	3,39	3,75	707	676	741	18,94	17,94	19,82	3796	3602	3965	18,94	17,94	19,82	3796	3602	3965
DPG1ls	0,36	0,34	0,39	61	59	65	1,76	1,65	1,85	312	296	330	7,02	6,58	7,50	1256	1178	1339	7,02	6,58	7,50	1256	1178	1339
DPG2	0,54	0,52	0,56	107	103	111	3,55	3,33	3,75	706	663	741	6,88	4,56	10,60	1372	908	2122	6,88	4,56	10,60	1372	908	2122
DPG2ls	0,36	0,33	0,38	61	59	65	1,29	0,82	1,77	227	143	314	1,86	1,39	2,53	325	243	443	1,86	1,39	2,53	325	243	443
DPG3	0,54	0,51	0,56	107	103	111	3,54	3,41	3,72	707	676	741	7,18	5,38	10,84	1435	1074	2168	7,18	5,38	10,84	1435	1074	2168
DPG3ls	0,36	0,34	0,38	61	59	65	1,37	0,92	1,75	243	163	314	2,16	1,57	2,85	384	280	510	2,16	1,57	2,85	384	280	510
PD1li	0,50	0,41	0,57	11	10	15	1,55	1,42	1,69	18	16	21	2,59	2,38	2,83	24	22	27	2,59	2,38	2,83	24	22	27
PD1lir	0,53	0,45	0,62	11	10	15	0,64	0,57	0,73	15	13	18	0,71	0,64	0,80	19	17	22	0,71	0,64	0,80	19	17	22
PD12i	0,72	0,65	0,81	10	10	11	2,12	1,98	2,29	17	16	17	3,49	3,30	3,72	23	22	24	3,49	3,30	3,72	23	22	24
PD13	5,09	4,63	5,60	10	9	10	8,40	7,85	9,11	16	15	17	11,69	11,15	12,28	22	22	23	11,69	11,15	12,28	22	22	23
PD2li	1,15	1,01	1,31	10	10	12	3,53	3,22	3,85	17	16	18	5,89	5,47	6,52	23	23	26	5,89	5,47	6,52	23	23	26
PD22i	1,58	1,43	1,83	10	10	11	4,44	4,13	4,85	17	16	18	7,50	6,60	8,23	23	22	24	7,50	6,60	8,23	23	22	24
PD23	10,12	9,51	11,22	10	10	11	16,32	15,41	17,21	17	16	17	22,47	21,68	23,68	23	23	24	22,47	21,68	23,68	23	23	24
PGls	1,00	0,93	1,08	271	254	293	4,52	4,20	4,91	1242	1152	1351	9,48	8,85	10,50	2608	2434	2893	9,48	8,85	10,50	2608	2434	2893

Tabuľka B.37: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 20% ($de = 0.2$).

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,98	0,91	1,05	55	54	56	1,22	1,13	1,29	76	74	78	1,50	1,41	1,56	101	97	106	1,50	1,41	1,56	101	97	106
APG	0,71	0,69	0,73	117	114	121	4,97	4,81	5,12	828	800	857	29,18	28,02	30,70	4877	4684	5132	29,18	28,02	30,70	4877	4684	5132
APGls	1,60	1,54	1,65	156	151	160	11,35	11,00	11,88	1130	1102	1171	70,35	67,30	73,84	7030	6721	7369	70,35	67,30	73,84	7030	6721	7369
DPG1	1,15	1,10	1,18	117	114	121	8,04	7,74	8,35	828	800	857	47,49	45,50	49,39	4898	4706	5089	47,49	45,50	49,39	4898	4706	5089
DPG1ls	0,74	0,70	0,78	67	64	71	3,95	3,79	4,17	372	354	390	17,99	17,02	18,75	1704	1610	1773	17,99	17,02	18,75	1704	1610	1773
DPG2	1,15	1,12	1,18	117	114	121	8,05	7,50	8,38	827	775	857	17,67	11,08	27,52	1817	1142	2823	17,67	11,08	27,52	1817	1142	2823
DPG2ls	0,74	0,70	0,79	67	64	71	3,26	2,22	4,04	304	203	380	4,76	3,57	6,15	443	328	574	4,76	3,57	6,15	443	328	574
DPG3	1,15	1,11	1,19	117	114	121	8,05	7,72	8,32	827	796	857	18,19	12,61	28,05	1873	1287	2884	18,19	12,61	28,05	1873	1287	2884
DPG3ls	0,74	0,71	0,79	67	64	71	3,45	2,37	4,03	324	223	380	5,55	3,82	6,78	524	362	641	5,55	3,82	6,78	524	362	641
PD11i	1,02	0,89	1,19	10	9	12	3,59	3,30	3,86	16	15	18	5,72	5,33	6,36	23	22	24	5,72	5,33	6,36	23	22	24
PD11ir	1,07	0,93	1,27	10	9	12	1,38	1,21	1,53	13	12	15	1,52	1,37	1,66	17	16	19	1,52	1,37	1,66	17	16	19
PD12i	1,32	1,18	1,61	9	9	10	4,45	4,07	4,76	16	15	17	7,02	6,55	7,47	22	22	23	7,02	6,55	7,47	22	22	23
PD13	4,67	4,01	4,90	9	8	9	7,99	7,61	8,59	15	15	16	11,39	10,67	11,85	22	21	22	11,39	10,67	11,85	22	21	22
PD21i	2,43	1,99	2,76	10	9	11	7,98	7,40	8,40	16	15	17	12,61	11,74	13,70	22	22	24	12,61	11,74	13,70	22	22	24
PD22i	3,05	2,59	3,47	10	9	12	9,25	8,63	9,98	17	16	19	14,86	13,88	15,97	23	22	25	14,86	13,88	15,97	23	22	25
PD23	9,71	8,97	10,35	10	9	10	15,62	14,99	16,61	16	16	17	21,96	20,96	23,07	23	22	23	21,96	20,96	23,07	23	22	23
PGLs	2,30	2,17	2,44	331	312	350	13,70	12,48	14,85	1997	1821	2161	31,83	29,29	35,28	4646	4289	5142	31,83	29,29	35,28	4646	4289	5142

Tabuľka B.38: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 40% ($de = 0.4$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		# iterácií	
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	1,70	1,63	1,77	71	70	73	2,19	2,08	2,27	98	93	103	3,12	2,99	3,25	148	143	158
APG	1,76	1,68	1,82	131	125	135	13,14	12,61	13,97	986	948	1034	89,69	85,13	95,11	6744	6396	7183
APGls	3,68			166			27,08			1251			190,65			8808		
DPG1	2,84			132			20,96			983			144,16			6749		
DPG1ls	1,77	1,70	1,83	74	71	76	10,57	10,17	11,07	460	442	482	56,12	54,02	58,92	2454	2368	2553
DPG2	2,83	2,69	2,92	131	125	135	21,15	20,24	22,51	986	948	1034	58,76	41,84	82,31	2741	1953	3852
DPG2ls	1,76	1,68	1,82	74	71	76	9,48	5,68	10,95	410	242	471	15,43	10,14	22,59	666	432	982
DPG3	2,83	2,68	2,97	131	125	135	21,14	20,30	22,57	986	948	1034	59,62	42,67	84,04	2785	1994	3935
DPG3ls	1,77	1,70	1,83	74	71	76	9,83	6,49	10,94	427	283	471	17,49	12,51	23,75	763	544	1039
PD1i	2,67	2,14	3,12	9	8	10	9,95	9,18	10,91	15	15	16	15,20	14,04	16,77	22	21	23
PD1ir	2,79	2,24	3,24	9	8	10	3,64	3,25	4,14	12	12	13	4,04	3,69	4,50	17	16	18
PD12i	3,06	2,43	3,54	9	8	10	10,81	10,09	11,82	15	15	16	16,55	15,39	17,87	22	21	22
PD13	4,32	4,00	4,90	8	8	9	7,85	7,60	8,13	15	15	15	10,98	10,62	11,61	21	21	22
PD21i	5,58	5,10	7,20	9	9	10	22,23	19,31	24,11	16	15	16	33,43	30,54	38,40	22	22	24
PD22i	6,39	5,63	7,56	9	9	10	22,69	21,06	24,36	16	15	17	34,76	32,11	38,32	22	22	23
PD23	9,07	8,45	9,92	9	9	10	15,41	14,27	16,12	16	15	16	21,44	20,37	22,80	22	22	23
PGls	6,49			436			56,88			3856			154,67			10488		

Tabuľka B.39: Vplyv riedkosti matice dát 2: Porovnanie metód riešenia na 50-tich úlohách (1.2) s hustotou zaplnenia matice dát 100% ($de = 1$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácii		čas		# iterácii		čas		# iterácii		čas		# iterácii		# iterácii	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	3,52	3,33	3,64	175	168	181	4,15	4,04	4,23	209	205	213	6,39	6,17	6,73	334	323	350
APG	1,99			152			17,45			1337			152,15			11661		
DPG1ls	1,82	1,75	1,92	85	81	88	12,95	12,18	13,81	622	582	664	94,17	89,16	98,39	4542	4296	4700
DPG2	3,04	2,94	3,16	153	148	158	26,72	25,30	27,65	1349	1274	1397	116,86	72,27	164,12	5902	3649	8350
DPG2ls	1,82	1,75	1,92	85	81	88	12,95	12,18	13,81	622	582	664	37,48	25,81	57,91	1801	1236	2811
DPG3	3,04	2,94	3,16	153	148	158	26,72	25,33	27,68	1349	1274	1397	119,19	73,64	167,69	6022	3722	8540
DPG3ls	1,83	1,75	2,17	85	81	88	12,96	12,17	13,81	622	582	664	40,36	30,44	60,60	1944	1465	2950
PD11i	3,06	2,61	3,97	8	8	9	14,32	12,65	17,32	15	14	15	20,52	18,47	24,00	21	21	24
PD11ir	3,15	2,70	4,08	8	8	9	4,62	3,87	5,58	12	11	12	4,97	4,21	5,96	16	15	17
PD12i	3,36	2,72	3,97	10	9	11	13,32	12,24	14,61	16	16	18	19,61	17,95	22,32	23	22	24
PD13	3,71	3,47	4,20	7	7	8	7,21	7,03	7,53	14	14	14	10,31	10,04	10,64	20	20	20
PD21i	7,64	6,02	9,40	9	8	10	31,84	29,76	37,28	15	14	16	45,64	40,92	53,00	22	21	24
PD22i	7,32	5,92	8,52	10	9	11	27,94	25,17	32,69	17	16	18	40,89	37,10	46,12	23	23	24
PD23	7,69	7,32	8,09	8	8	8	13,30	12,86	13,77	14	14	14	19,87	19,33	20,47	21	21	21

Tabuľka B.40: Vplyv vzdialenosti od optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) pri počiatkovej vzdialenosti od optimálneho riešenia $\rho = 0.1$.

Metóda	Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,59	0,53	0,66	37	36	37	37	0,97	0,90	1,06	94	90	98	3,03	2,89	3,24	406	380	431
APG	0,01	0,01	0,01	7	7	7	8	0,13	0,13	0,14	89	86	92	0,88	0,84	0,91	608	585	635
APGls	0,03	0,03	0,03	7	7	7	7	0,24	0,22	0,25	84	81	88	1,57	1,48	1,67	572	551	603
DPG1	0,02	0,02	0,03	7	7	7	8	0,23	0,21	0,23	89	86	92	1,52	1,45	1,58	608	586	635
DPG1ls	0,03	0,02	0,03	6	6	6	6	0,12	0,12	0,14	41	39	44	0,63	0,58	0,70	224	212	240
DPG2	0,02	0,02	0,02	7	7	7	8	0,23	0,22	0,24	89	86	92	1,50	1,28	1,59	602	516	635
DPG2ls	0,03	0,03	0,03	6	6	6	6	0,12	0,11	0,13	41	39	44	0,32	0,28	0,41	106	92	139
DPG3	0,02	0,02	0,02	7	7	7	8	0,23	0,22	0,23	89	86	92	1,50	1,34	1,58	604	539	635
DPG3ls	0,03	0,03	0,03	6	6	6	6	0,12	0,12	0,14	41	39	44	0,34	0,31	0,44	119	111	152
PD1li	0,14	0,12	0,16	10	10	10	11	0,61	0,54	0,67	17	17	18	1,20	1,10	1,31	24	23	25
PD1lir	0,16	0,14	0,18	10	10	10	11	0,20	0,18	0,23	13	13	14	0,23	0,21	0,27	18	17	19
PD12i	0,23	0,21	0,24	10	10	10	10	1,03	0,96	1,10	17	17	17	1,93	1,84	2,07	23	23	24
PD13	5,63	5,49	5,92	11	11	11	11	8,70	8,56	9,04	17	17	17	11,85	11,59	12,40	23	23	24
PD2li	0,38	0,33	0,47	11	11	11	13	1,50	1,42	1,59	18	18	19	2,77	2,57	3,35	24	24	29
PD22i	0,57	0,54	0,63	11	11	11	12	2,07	1,97	2,32	17	17	18	4,26	4,00	4,55	24	24	25
PD23	10,48	10,17	10,74	11	11	11	11	16,57	15,69	17,41	18	17	18	22,68	22,19	23,26	24	24	24
PGls	0,02	0,02	0,03	8	7	8	8	0,23	0,20	0,26	124	111	138	1,20	1,07	1,30	662	580	728

Tabuľka B.41: Vplyv vzdialenosti od optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) pri počiatkovej vzdialenosti od optimálneho riešenia $\rho = 1$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	1,38	1,30	1,47	156	152	160	1,54	1,45	1,63	180	175	185	1,70	1,62	1,79	204	199	212
	APG	0,06	0,05	0,06	38	36	38	0,37	0,36	0,39	260	252	271	2,12	2,00	2,24	1507	1421	1593
	APGls	0,13	0,10	0,15	47	35	53	0,88	0,61	1,02	331	243	376	5,45	3,82	6,36	1992	1399	2302
	DPG1	0,10	0,09	0,10	38	36	38	0,64	0,62	0,67	260	252	271	3,67	3,46	3,86	1509	1432	1583
	DPG1ls	0,08	0,07	0,08	24	21	26	0,33	0,29	0,37	117	106	126	1,44	1,34	1,54	513	469	551
	DPG2	0,10	0,09	0,10	38	36	38	0,64	0,62	0,67	260	252	271	2,42	1,52	3,67	993	616	1502
	DPG2ls	0,08	0,06	0,08	24	21	26	0,25	0,22	0,34	85	73	119	0,44	0,38	0,53	152	137	180
	DPG3	0,10	0,09	0,10	38	36	38	0,64	0,62	0,67	260	252	271	2,48	1,52	3,70	1021	622	1506
	DPG3ls	0,08	0,06	0,08	24	21	26	0,27	0,24	0,33	95	83	119	0,52	0,46	0,65	186	164	221
	PD11i	0,22	0,18	0,26	11	10	12	0,70	0,63	0,76	17	16	18	1,24	1,17	1,45	24	23	27
	PD11ir	0,23	0,21	0,26	11	10	12	0,34	0,29	0,38	14	13	15	0,37	0,31	0,41	18	17	19
	PD12i	0,34	0,31	0,39	10	10	11	1,11	1,01	1,22	17	16	17	1,94	1,82	2,05	23	23	23
	PD13	5,55	5,43	5,77	11	11	11	8,78	8,39	9,43	17	17	18	12,08	11,87	12,44	24	24	24
	PD21i	0,54	0,51	0,63	11	11	12	1,69	1,55	1,83	17	17	18	2,97	2,79	3,35	24	24	27
	PD22i	0,83	0,72	0,93	12	11	12	2,41	2,18	2,75	18	17	19	4,22	3,89	4,48	24	24	25
	PD23	10,65	10,00	11,88	11	11	12	16,87	16,32	17,60	18	18	18	22,44	21,72	24,23	24	24	25
	PGls	0,09	0,08	0,10	47	41	50	0,70	0,65	0,75	393	354	425	1,87	1,77	2,03	1065	999	1151

Tabuľka B.42: Vplyv vzdialenosti od optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) pri počiatkovej vzdialenosti od optimálneho riešenia $\rho = 10$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}								
	čas			# iterácii			čas			# iterácii			čas			# iterácii					
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,64	0,58	0,71	44	42	46	0,80	0,74	0,88	68	64	71	0,96	0,89	1,04	93	89	97			
APG	0,14	0,13	0,15	98	94	101	0,85	0,82	0,89	608	583	630	4,07	3,88	4,32	2907	2762	3068			
APGls	0,35	0,32	0,38	132	126	138	2,34	2,21	2,48	865	821	921	11,93	11,14	13,09	4376	4087	4816			
DPG1	0,24	0,23	0,25	98	94	101	1,48	1,43	1,53	608	583	630	7,04	6,63	7,49	2917	2746	3095			
DPG1ls	0,17	0,16	0,18	57	54	58	0,73	0,65	0,79	263	228	276	2,63	2,49	2,79	923	871	992			
DPG2	0,24	0,23	0,25	98	94	101	1,46	1,17	1,53	599	477	630	2,52	1,66	3,82	1040	672	1572			
DPG2ls	0,17	0,15	0,18	57	54	58	0,39	0,34	0,52	137	119	188	0,58	0,52	0,69	207	186	253			
DPG3	0,24	0,23	0,25	98	94	101	1,46	1,22	1,54	601	502	630	2,67	1,87	3,89	1105	769	1608			
DPG3ls	0,17	0,15	0,18	57	54	58	0,42	0,37	0,54	148	133	204	0,70	0,58	1,01	252	212	375			
PD11i	0,23	0,19	0,27	12	10	15	0,64	0,57	0,70	18	17	21	1,12	1,03	1,24	25	23	28			
PD11ir	0,25	0,21	0,29	12	10	15	0,30	0,26	0,34	15	14	18	0,33	0,29	0,37	20	18	23			
PD12i	0,39	0,34	0,45	10	10	11	1,06	0,96	1,14	17	16	18	1,79	1,64	1,92	23	23	24			
PD13	5,22	4,96	5,82	10	10	11	8,50	7,99	8,96	17	16	18	11,65	11,42	12,34	23	23	24			
PD21i	0,57	0,49	0,64	11	10	12	1,59	1,49	1,72	18	17	19	2,76	2,53	2,93	24	23	26			
PD22i	0,88	0,79	0,97	11	10	12	2,29	2,06	2,45	18	16	19	3,89	3,57	4,31	24	23	25			
PD23	10,18	9,34	10,80	11	10	11	15,99	14,99	17,35	17	16	18	22,00	21,15	23,13	24	23	24			
PGLs	0,40	0,34	0,44	225	204	245	1,41	1,29	1,52	803	746	875	2,67	2,48	2,91	1530	1424	1630			

Tabuľka B.43: Vplyv vzdialenosti od optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) pri počiatkovej vzdialenosti od optimálneho riešenia $\rho = 100$.

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,47	0,43	0,56	21	20	22	0,60	0,56	0,71	40	38	44	0,78	0,72	0,88	67	63	71	0,78	0,72	0,88	67	63	71
APG	0,32	0,30	0,34	226	214	239	1,78	1,67	1,88	1261	1185	1346	7,09	6,65	7,44	5045	4732	5282	7,09	6,65	7,44	5045	4732	5282
APGls	0,82	0,75	0,88	306	291	321	4,93	4,59	5,27	1800	1700	1920	21,27	19,68	22,71	7751	7179	8271	21,27	19,68	22,71	7751	7179	8271
DPG1	0,55	0,52	0,59	226	214	239	3,07	2,93	3,27	1263	1216	1350	12,41	11,75	12,93	5119	4851	5317	12,41	11,75	12,93	5119	4851	5317
DPG1ls	0,39	0,36	0,44	140	131	148	1,55	1,47	1,66	551	528	584	4,43	4,20	4,62	1548	1475	1618	4,43	4,20	4,62	1548	1475	1618
DPG2	0,55	0,52	0,58	226	214	239	2,31	1,63	3,20	947	665	1311	2,79	2,09	3,96	1140	850	1625	2,79	2,09	3,96	1140	850	1625
DPG2ls	0,39	0,36	0,42	140	131	148	0,66	0,59	0,80	234	215	291	0,86	0,75	1,01	305	277	370	0,86	0,75	1,01	305	277	370
DPG3	0,55	0,52	0,58	226	214	239	2,37	1,69	3,20	974	698	1311	3,03	2,19	4,13	1244	901	1693	3,03	2,19	4,13	1244	901	1693
DPG3ls	0,39	0,36	0,43	140	131	148	0,72	0,66	0,85	263	243	308	1,08	0,86	1,45	389	321	509	1,08	0,86	1,45	389	321	509
PD1i	0,29	0,25	0,34	17	15	19	0,65	0,59	0,70	24	21	26	1,14	1,03	1,26	30	27	34	1,14	1,03	1,26	30	27	34
PD1lr	0,30	0,27	0,34	17	14	19	0,37	0,33	0,41	21	19	24	0,40	0,36	0,45	26	23	28	0,40	0,36	0,45	26	23	28
PD12i	0,60	0,55	0,69	17	16	18	1,21	1,10	1,31	23	22	25	1,94	1,84	2,10	30	29	31	1,94	1,84	2,10	30	29	31
PD13	6,57	5,98	7,07	13	12	14	9,72	9,37	10,10	19	19	20	12,99	12,45	13,60	26	25	27	12,99	12,45	13,60	26	25	27
PD21i	0,68	0,61	0,75	18	16	21	1,54	1,44	1,65	24	22	27	2,71	2,58	2,89	31	28	34	2,71	2,58	2,89	31	28	34
PD22i	1,29	1,17	1,38	18	17	19	2,53	2,36	2,72	24	23	26	4,16	3,89	4,49	31	30	32	4,16	3,89	4,49	31	30	32
PD23	11,74	11,07	12,47	12	12	13	17,86	16,88	18,80	19	18	20	23,74	23,18	24,57	25	25	26	23,74	23,18	24,57	25	25	26
PGls	1,92	1,77	2,10	1101	1010	1216	3,47	3,23	3,71	1993	1849	2156	4,79	4,45	5,07	2752	2575	2922	4,79	4,45	5,07	2752	2575	2922

Tabuľka B.44: Vplyv vzdialenosti od optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) pri počiatkoch vzdialenosti od optimálneho riešenia $\rho = 1000$.

Presn. ϵ	Metóda	10^{-3}			10^{-6}			10^{-9}											
		čas	# iterácii		čas	# iterácii		čas	# iterácii										
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max			
	ADMM	0,95	0,87	1,09	92	82	103	1,14	1,06	1,29	120	108	131	1,36	1,28	1,51	153	143	162
	APG	0,91	0,87	0,95	643	614	672	3,09	2,89	3,23	2188	2058	2286	10,32	9,72	10,78	7312	6867	7578
	APGls	2,41	2,27	2,54	881	845	925	8,36	7,92	8,85	3037	2864	3219	30,48	28,73	32,28	11050	10391	11747
	DPG1	1,57	1,50	1,64	643	614	672	5,36	5,13	5,57	2194	2110	2283	18,13	17,00	18,95	7452	7022	7786
	DPG1ls	1,19	1,11	1,31	423	402	444	3,07	2,92	3,23	1067	1023	1113	6,90	6,56	7,25	2386	2272	2496
	DPG2	1,54	1,40	1,65	628	573	672	2,78	2,04	4,76	1137	827	1958	3,30	2,53	5,27	1345	1037	2167
	DPG2ls	1,13	1,05	1,21	402	384	425	1,48	1,37	1,55	521	497	545	1,70	1,58	1,78	595	569	615
	DPG3	1,55	1,44	1,65	632	587	672	2,92	2,16	4,85	1192	880	1987	3,54	2,63	5,41	1449	1073	2217
	DPG3ls	1,19	1,05	1,33	425	392	457	1,54	1,42	1,67	548	523	577	1,88	1,71	2,20	664	607	775
	PD11i	0,54	0,49	0,60	22	20	24	0,81	0,75	0,86	27	26	29	1,30	1,22	1,38	34	32	36
	PD11ir	0,58	0,52	0,65	22	20	24	0,65	0,61	0,71	26	24	28	0,69	0,65	0,75	30	29	32
	PD12i	1,27	1,14	1,36	24	22	25	1,73	1,58	1,83	29	27	31	2,44	2,32	2,57	36	34	37
	PD13	8,00	7,45	8,55	16	15	17	11,18	10,43	11,82	22	21	23	14,48	13,90	14,96	29	28	30
	PD21i	1,18	1,08	1,28	22	20	24	1,86	1,74	1,98	28	26	30	2,98	2,83	3,23	34	32	36
	PD22i	3,09	2,89	3,29	30	28	32	4,15	3,88	4,44	36	33	38	5,68	5,39	6,08	42	40	44
	PD23	13,83	12,80	15,67	14	13	16	20,07	18,94	21,35	21	20	22	26,19	25,11	27,12	28	27	29
	PGLs	12,30	10,92	13,74	7038	6253	7832	15,09	13,47	16,51	8633	7708	9457	16,48	14,90	17,87	9428	8546	10231

Tabuľka B.45: Vplyv vzdialenosti od optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) pri počiatkoch vzdialenosti od optimálneho riešenia $\rho = 10000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	4,61	4,06	5,03	643	566	716	4,98	4,40	5,47	699	619	783	5,24	4,67	5,70	738	660	818
APG	2,92	2,82	3,02	2071	2011	2130	5,30	5,07	5,48	3757	3601	3880	13,34	12,54	14,38	9465	8936	10,2K
APGls	7,95	7,66	8,30	2893	2797	3030	14,40	13,90	15,09	5231	5068	5465	38,96	35,81	42,01	14,1K	13,0K	15,3K
DPG1	5,03	4,86	5,22	2070	2011	2131	9,19	8,77	9,53	3783	3622	3902	23,68	21,99	24,86	9763	9106	10,2K
DPG1ls	4,25	4,09	4,42	1475	1435	1535	6,48	6,26	6,69	2244	2188	2307	11,04	10,51	11,41	3813	3653	3937
DPG2	4,82	4,42	5,11	1978	1807	2087	5,84	5,21	6,69	2396	2129	2730	6,39	5,74	7,24	2619	2339	2948
DPG2ls	3,95	3,74	4,17	1370	1317	1442	4,53	4,33	4,78	1566	1496	1646	4,77	4,57	5,01	1645	1577	1723
DPG3	4,72	4,30	5,11	1941	1765	2087	5,78	5,12	6,78	2377	2095	2764	6,58	5,69	7,66	2707	2354	3140
DPG3ls	4,20	3,90	4,51	1464	1368	1573	4,72	4,40	5,06	1641	1539	1776	5,03	4,72	5,36	1749	1645	1870
PD11i	1,11	1,02	1,21	27	25	29	1,26	1,19	1,33	30	28	31	1,70	1,59	1,78	36	35	38
PD11ir	1,12	1,01	1,23	27	25	29	1,19	1,08	1,27	29	27	30	1,23	1,12	1,32	33	32	35
PD12i	2,54	2,36	2,77	30	28	33	2,76	2,55	2,95	33	30	35	3,44	3,28	3,69	39	37	42
PD13	10,42	9,66	11,25	21	19	22	12,51	11,81	13,29	25	23	26	15,77	15,04	16,44	31	30	32
PD21i	2,29	2,07	2,69	26	24	32	2,66	2,50	3,02	29	28	35	3,69	3,47	4,13	36	34	42
PD22i	5,96	5,68	6,31	34	32	36	6,51	6,19	6,80	37	35	39	7,95	7,63	8,40	44	41	46
PD23	17,92	15,91	19,50	19	17	20	22,62	20,61	24,31	23	22	25	28,80	27,23	30,23	30	29	31
PGLs	106			61,5K			113			65,5K			114			66,3K		

Tabuľka B.46: Vplyv počtu nenulových prvkov optimálneho riešenia na 50-tich úlohách (1.2) s optimálnym riešením s počtom nenulových prvkov $pn = 100$.

Metóda	Presn. ϵ			10^{-3}			10^{-6}			10^{-9}		
	čas			# iterácií			čas			čas		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,74	0,69	0,83	62	59	64	1,13	1,05	1,23	121	114	127
APG	0,13	0,12	0,16	94	87	109	0,67	0,63	0,70	475	448	502
APGls	0,36	0,25	0,42	133	89	153	1,92	1,27	2,13	712	478	777
DPG1	0,22	0,21	0,25	89	86	98	1,20	1,12	1,26	492	462	522
DPG1ls	0,12	0,10	0,13	39	34	42	0,20	0,19	0,23	71	66	78
DPG2	0,20	0,20	0,22	82	78	87	0,39	0,35	0,60	155	138	243
DPG2ls	0,12	0,10	0,13	38	33	41	0,16	0,14	0,19	53	48	66
DPG3	0,22	0,20	0,23	88	83	94	0,43	0,41	0,45	173	168	180
DPG3ls	0,12	0,10	0,12	39	34	41	0,15	0,14	0,19	51	47	64
PD11i	0,21	0,18	0,26	15	14	18	0,43	0,40	0,47	22	21	25
PD11ir	0,22	0,19	0,27	15	13	18	0,23	0,20	0,28	19	17	22
PD12i	0,36	0,32	0,48	15	14	16	0,71	0,67	0,80	21	20	23
PD13	7,64	7,01	8,57	15	14	17	10,90	9,99	11,71	21	20	23
PD21i	0,53	0,46	0,61	16	14	17	1,03	0,96	1,11	22	21	24
PD22i	0,78	0,69	0,90	14	13	16	1,54	1,46	1,65	21	20	22
PD23	13,64	12,12	15,04	14	13	16	19,60	18,45	21,00	21	20	22
PGls	0,16	0,14	0,18	88	79	98	0,20	0,17	0,21	109	100	121
										0,22	0,19	0,24
										122	115	135

Tabuľka B.47: Vplyv počtu nenulových prvkov optimálneho riešenia na 50-tich úlohách (1.2) s optimálnym riešením s počtom nenulových prvkov $pn = 300$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií			Metóda	čas		# iterácií			čas	pr		# iterácií			pr
	pr	min	max	pr	min		pr	min	max	pr	min		pr	min	max	pr	min	max
	0,76	0,69	0,84	62	60	66	1,29	1,19	1,43	144	135	154	1,91	1,77	2,08	238	222	253
ADMM																		
APG	0,12	0,11	0,12	82	80	86	0,71	0,67	0,76	501	474	533	2,32	2,17	2,52	1646	1550	1775
APGls	0,31	0,28	0,32	113	109	118	2,03	1,89	2,16	752	707	796	7,29	6,55	7,88	2667	2427	2864
DPG1	0,20	0,20	0,21	82	80	86	1,23	1,16	1,29	503	477	531	4,04	3,76	4,34	1661	1556	1775
DPG1ls	0,13	0,11	0,14	43	39	48	0,31	0,28	0,34	108	96	120	0,56	0,51	0,61	201	185	217
DPG2	0,20	0,20	0,21	82	80	86	0,60	0,40	1,25	244	163	510	0,84	0,57	1,61	337	227	661
DPG2ls	0,13	0,11	0,15	43	40	47	0,20	0,17	0,23	68	58	79	0,25	0,22	0,29	83	73	97
DPG3	0,20	0,20	0,21	82	80	86	0,68	0,49	1,24	277	199	510	1,17	0,71	1,88	476	288	775
DPG3ls	0,13	0,11	0,14	43	41	47	0,21	0,19	0,25	72	62	84	0,27	0,24	0,31	93	79	111
PD11i	0,20	0,17	0,24	14	12	18	0,44	0,40	0,50	21	19	24	0,69	0,61	0,75	27	25	31
PD11ir	0,23	0,20	0,27	14	12	18	0,23	0,21	0,27	18	16	21	0,24	0,21	0,28	22	20	26
PD12i	0,35	0,32	0,39	12	12	13	0,72	0,66	0,77	19	18	19	1,13	1,04	1,23	25	25	26
PD13	6,27	5,98	6,69	12	12	13	9,50	8,95	10,21	19	18	20	12,73	12,21	13,36	25	24	26
PD21i	0,49	0,44	0,55	13	12	14	1,07	1,01	1,18	19	19	21	1,67	1,49	1,83	26	25	27
PD22i	0,79	0,73	0,85	13	12	14	1,57	1,48	1,71	19	18	20	2,47	2,27	2,63	26	24	27
PD23	11,36	10,32	12,38	12	11	13	17,38	16,75	18,25	18	18	19	23,28	22,23	24,22	25	24	26
PGls	0,21	0,19	0,23	118	110	126	0,30	0,28	0,34	170	160	179	0,39	0,36	0,43	217	205	232

Tabuľka B.48: Vplyv počtu nenulových prvkov optimálneho riešenia na 50-tich úlohách (1.2) s optimálnym riešením s počtom nenulových prvkov $pn = 600$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,80	0,75	0,90	70	67	74	1,11	1,05	1,22	117	111	124	1,41	1,34	1,54	163	155	175
	APG	0,13	0,12	0,13	88	84	91	0,75	0,71	0,79	534	500	556	2,82	2,63	2,98	2003	1871	2123
	APGls	0,32	0,29	0,35	120	114	125	2,10	1,93	2,23	780	720	826	8,63	8,10	9,07	3158	2961	3344
	DPG1	0,22	0,21	0,23	88	84	91	1,31	1,23	1,36	536	500	556	4,91	4,67	5,17	2022	1913	2127
	DPG1ls	0,14	0,13	0,16	48	44	52	0,46	0,42	0,53	167	150	181	1,08	0,99	1,20	393	363	415
	DPG2	0,22	0,21	0,23	88	84	91	1,19	0,86	1,37	488	348	556	1,64	1,14	2,55	668	459	1040
	DPG2ls	0,14	0,13	0,16	48	44	52	0,24	0,21	0,32	82	71	109	0,32	0,29	0,41	109	95	139
	DPG3	0,22	0,21	0,23	88	84	91	1,21	0,92	1,36	496	373	556	1,70	1,19	2,60	698	484	1064
	DPG3ls	0,14	0,13	0,15	48	44	52	0,25	0,23	0,33	89	79	119	0,36	0,32	0,42	126	114	151
	PD11i	0,21	0,18	0,24	13	11	16	0,49	0,45	0,57	20	17	23	0,82	0,73	0,95	26	24	31
	PD11ir	0,23	0,20	0,28	13	11	16	0,25	0,22	0,29	17	14	20	0,26	0,23	0,30	21	19	24
	PD12i	0,35	0,31	0,39	11	11	12	0,81	0,75	0,88	18	17	19	1,33	1,24	1,46	24	24	25
	PD13	5,77	5,48	6,31	11	11	12	8,99	8,45	9,35	18	17	18	12,26	11,98	12,93	24	24	25
	PD21i	0,52	0,47	0,57	12	11	13	1,21	1,10	1,32	18	17	19	1,98	1,86	2,16	25	24	26
	PD22i	0,80	0,72	0,90	12	11	13	1,78	1,65	1,93	19	18	20	2,89	2,69	3,07	25	24	26
	PD23	10,87	10,15	11,62	12	11	12	16,85	15,89	17,96	18	17	19	22,76	22,13	23,56	24	24	25
	PGls	0,28	0,25	0,31	155	142	168	0,52	0,46	0,59	293	272	318	0,76	0,69	0,83	432	405	464

Tabuľka B.49: Vplyv počtu nenulových prvkov optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) s optimálnym riešením s počtom nenulových prvkov $pn = 1200$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		# iterácií	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,64	0,58	0,71	44	42	46	0,80	0,74	0,88	68	64	71	0,96	0,89	1,04	93	89	97
	APG	0,14	0,13	0,15	98	94	101	0,85	0,82	0,89	608	583	630	4,07	3,88	4,32	2907	2762	3068
	APGls	0,35	0,32	0,38	132	126	138	2,34	2,21	2,48	865	821	921	11,93	11,14	13,09	4376	4087	4816
	DPG1	0,24	0,23	0,25	98	94	101	1,48	1,43	1,53	608	583	630	7,04	6,63	7,49	2917	2746	3095
	DPG1ls	0,17	0,16	0,18	57	54	58	0,73	0,65	0,79	263	228	276	2,63	2,49	2,79	923	871	992
	DPG2	0,24	0,23	0,25	98	94	101	1,46	1,17	1,53	599	477	630	2,52	1,66	3,82	1040	672	1572
	DPG2ls	0,17	0,15	0,18	57	54	58	0,39	0,34	0,52	137	119	188	0,58	0,52	0,69	207	186	253
	DPG3	0,24	0,23	0,25	98	94	101	1,46	1,22	1,54	601	502	630	2,67	1,87	3,89	1105	769	1608
	DPG3ls	0,17	0,15	0,18	57	54	58	0,42	0,37	0,54	148	133	204	0,70	0,58	1,01	252	212	375
	PD11i	0,23	0,19	0,27	12	10	15	0,64	0,57	0,70	18	17	21	1,12	1,03	1,24	25	23	28
	PD11ir	0,25	0,21	0,29	12	10	15	0,30	0,26	0,34	15	14	18	0,33	0,29	0,37	20	18	23
	PD12i	0,39	0,34	0,45	10	10	11	1,06	0,96	1,14	17	16	18	1,79	1,64	1,92	23	23	24
	PD13	5,22	4,96	5,82	10	10	11	8,50	7,99	8,96	17	16	18	11,65	11,42	12,34	23	23	24
	PD21i	0,57	0,49	0,64	11	10	12	1,59	1,49	1,72	18	17	19	2,76	2,53	2,93	24	23	26
	PD22i	0,88	0,79	0,97	11	10	12	2,29	2,06	2,45	18	16	19	3,89	3,57	4,31	24	23	25
	PD23	10,18	9,34	10,80	11	10	11	15,99	14,99	17,35	17	16	18	22,00	21,15	23,13	24	23	24
	PGLs	0,40	0,34	0,44	225	204	245	1,41	1,29	1,52	803	746	875	2,67	2,48	2,91	1530	1424	1630

Tabuľka B.50: Vplyv počtu nenulových prvkov optimálneho riešenia na 50-tich úlohách (1.2) s optimálnym riešením s počtom nenulových prvkov $pn = 1800$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,63	0,57	0,70	43	42	45	0,87	0,79	0,94	79	73	83	1,32	1,17	1,41	147	133	161
APG	0,15	0,14	0,15	101	99	106	0,98	0,95	1,05	697	674	734	6,00	5,75	6,32	4262	4085	4507
APGls	0,38	0,35	0,40	137	133	143	2,65	2,55	2,80	971	934	1021	16,86	15,86	17,77	6150	5769	6495
DPG1	0,25	0,24	0,27	101	99	106	1,70	1,65	1,80	697	674	734	10,35	9,88	10,91	4273	4107	4518
DPG1ls	0,18	0,17	0,20	62	60	66	0,97	0,89	1,02	348	330	365	5,15	4,83	5,54	1794	1688	1928
DPG2	0,25	0,24	0,26	101	99	106	1,70	1,65	1,79	697	674	734	4,20	3,07	5,83	1725	1267	2395
DPG2ls	0,18	0,17	0,20	62	60	66	0,80	0,62	1,01	285	213	356	1,37	1,07	1,81	480	393	634
DPG3	0,25	0,24	0,26	101	99	106	1,70	1,64	1,80	697	674	734	4,27	3,27	5,95	1756	1353	2457
DPG3ls	0,18	0,17	0,20	62	60	66	0,83	0,65	1,01	301	237	356	1,54	1,25	1,87	549	450	671
PD11i	0,26	0,21	0,31	11	10	13	0,99	0,90	1,11	18	16	20	1,93	1,79	2,24	24	23	28
PD11ir	0,29	0,23	0,34	11	10	13	0,39	0,35	0,42	15	14	17	0,47	0,43	0,50	19	18	21
PD12i	0,44	0,38	0,51	10	10	11	1,54	1,42	1,65	17	16	18	2,79	2,53	3,07	23	23	24
PD13	5,05	4,53	5,60	10	9	11	8,26	7,98	8,83	16	16	17	11,50	10,95	11,99	23	22	23
PD21i	0,67	0,56	0,75	11	10	14	2,46	2,24	2,66	17	16	21	4,74	4,37	5,44	24	23	27
PD22i	0,98	0,87	1,11	11	10	12	3,28	3,01	3,46	18	17	18	5,99	5,43	6,47	24	23	25
PD23	9,69	9,29	10,88	10	10	11	15,79	14,93	16,70	17	16	17	21,93	20,66	23,28	23	22	24
PGLs	0,50	0,46	0,55	281	262	309	4,47	4,04	5,04	2565	2323	2880	12,43	11,04	14,16	7132	6335	8080

Tabuľka B.51: Vplyv počtu nenulových prvkov optimálneho riešenia na 50-tich úlohách (1.2) s optimálnym riešením s počtom nenulových prvkov $pn = 2400$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií			
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	0,57	0,51	0,63	33	31	34	2,44	2,04	2,78	319	264	379	85,40	54,40	168,11	13,0K	8180	25,6K
APG	0,14	0,14	0,15	99	95	103	1,28	1,19	1,43	907	846	1001	10,57	9,56	11,56	7524	6798	8140
APGls	0,36	0,33	0,39	134	128	139	3,37	3,16	3,69	1237	1169	1346	28,18	25,84	31,13	10,3K	9454	11,3K
DPG1	0,25	0,24	0,26	99	95	103	2,21	2,05	2,46	907	846	1001	18,21	16,54	19,79	7519	6802	8131
DPG1ls	0,19	0,18	0,21	65	62	69	1,39	1,27	1,56	491	462	544	10,90	9,69	12,12	3788	3338	4177
DPG2	0,25	0,23	0,26	99	95	103	2,21	2,06	2,46	907	846	1001	16,14	12,52	19,48	6659	5150	8014
DPG2ls	0,19	0,17	0,21	65	62	69	1,38	1,26	1,51	491	462	544	9,93	7,74	12,21	3449	2682	4258
DPG3	0,25	0,23	0,26	99	95	103	2,21	2,05	2,44	907	846	1001	16,42	12,89	20,07	6780	5305	8292
DPG3ls	0,19	0,18	0,21	65	62	69	1,38	1,29	1,58	491	462	544	10,18	8,22	13,47	3543	2861	4681
PD11i	0,32	0,23	0,38	11	9	13	4,16	3,34	5,20	17	16	20	38,32	29,46	46,76	24	22	26
PD11ir	0,34	0,25	0,40	11	9	13	0,67	0,59	0,75	15	13	17	1,66	1,35	2,11	19	17	21
PD12i	0,49	0,41	0,56	11	10	12	4,39	3,38	5,16	20	19	21	30,08	21,57	46,03	27	26	29
PD13	4,62	4,48	5,24	9	9	10	8,03	7,55	8,34	16	15	16	11,25	10,59	11,86	22	21	23
PD21i	0,73			10			10,59			17			88,87			23		
PD22i	1,10			11			8,06			20			66,71			27		
PD23	9,11	8,44	9,77	9	9	10	15,33	14,14	16,38	16	15	17	21,34	20,73	22,16	22	22	23
PGls	0,53			305			22,03			12,9K			1062			0,62M		

Tabuľka B.52: Vplyv dynamického rozsahu optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) s optimálnym riešením s dynamickým rozsahom $dr = 1$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,59	0,53	0,67	37	35	40	0,76	0,68	0,84	62	57	68	0,92	0,84	1,02	87	82	95
	APG	0,12	0,12	0,13	86	83	90	0,72	0,68	0,75	507	483	523	3,35	3,12	3,54	2379	2232	2524
	APGls	0,32	0,30	0,34	117	113	122	1,98	1,90	2,08	729	689	766	9,87	9,16	10,59	3597	3353	3863
	DPG1	0,21	0,20	0,23	86	83	90	1,24	1,17	1,29	507	483	523	5,79	5,41	6,16	2381	2254	2522
	DPG1ls	0,16	0,15	0,17	53	50	56	0,63	0,58	0,67	223	210	238	2,24	2,03	2,34	785	729	832
	DPG2	0,21	0,21	0,22	86	83	90	1,24	1,17	1,28	507	483	523	2,52	1,66	3,57	1030	673	1455
	DPG2ls	0,16	0,14	0,17	53	50	56	0,36	0,31	0,44	121	110	150	0,55	0,49	0,62	191	173	215
	DPG3	0,21	0,21	0,22	86	83	90	1,24	1,17	1,28	507	483	523	2,60	1,92	3,61	1065	784	1489
	DPG3ls	0,16	0,15	0,17	53	50	56	0,37	0,35	0,45	132	122	166	0,64	0,55	0,93	228	198	319
	PD11i	0,22	0,18	0,26	11	9	14	0,64	0,59	0,72	18	16	21	1,13	1,03	1,32	24	22	27
	PD11ir	0,24	0,20	0,27	11	9	14	0,30	0,26	0,34	15	13	18	0,33	0,29	0,37	19	17	22
	PD12i	0,37	0,33	0,41	10	9	11	1,03	0,94	1,11	16	15	17	1,76	1,59	1,89	23	22	24
	PD13	4,71	4,49	5,22	9	9	10	8,04	7,56	8,36	16	15	16	11,22	10,95	11,68	22	22	23
	PD21i	0,55	0,48	0,62	10	10	12	1,53	1,40	1,75	16	16	19	2,68	2,51	2,86	23	22	25
	PD22i	0,88	0,73	0,97	11	9	11	2,22	2,07	2,42	17	16	18	3,82	3,36	4,15	23	22	24
	PD23	9,54	8,59	9,93	10	9	10	15,27	14,83	16,40	16	16	17	21,15	20,46	22,24	22	22	23
	PGls	0,34	0,31	0,37	193	177	207	1,12	1,01	1,22	643	572	697	2,28	2,04	2,46	1309	1173	1415

Tabuľka B.53: Vplyv dynamického rozsahu optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) s optimálnym riešením s dynamickým rozsahom $dr = 1000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		# iterácií	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	ADMM	0,64	0,58	0,71	44	42	46	0,80	0,74	0,88	68	64	71	0,96	0,89	1,04	93	89	97
	APG	0,14	0,13	0,15	98	94	101	0,85	0,82	0,89	608	583	630	4,07	3,88	4,32	2907	2762	3068
	APGls	0,35	0,32	0,38	132	126	138	2,34	2,21	2,48	865	821	921	11,93	11,14	13,09	4376	4087	4816
	DPG1	0,24	0,23	0,25	98	94	101	1,48	1,43	1,53	608	583	630	7,04	6,63	7,49	2917	2746	3095
	DPG1ls	0,17	0,16	0,18	57	54	58	0,73	0,65	0,79	263	228	276	2,63	2,49	2,79	923	871	992
	DPG2	0,24	0,23	0,25	98	94	101	1,46	1,17	1,53	599	477	630	2,52	1,66	3,82	1040	672	1572
	DPG2ls	0,17	0,15	0,18	57	54	58	0,39	0,34	0,52	137	119	188	0,58	0,52	0,69	207	186	253
	DPG3	0,24	0,23	0,25	98	94	101	1,46	1,22	1,54	601	502	630	2,67	1,87	3,89	1105	769	1608
	DPG3ls	0,17	0,15	0,18	57	54	58	0,42	0,37	0,54	148	133	204	0,70	0,58	1,01	252	212	375
	PD11i	0,23	0,19	0,27	12	10	15	0,64	0,57	0,70	18	17	21	1,12	1,03	1,24	25	23	28
	PD11ir	0,25	0,21	0,29	12	10	15	0,30	0,26	0,34	15	14	18	0,33	0,29	0,37	20	18	23
	PD12i	0,39	0,34	0,45	10	10	11	1,06	0,96	1,14	17	16	18	1,79	1,64	1,92	23	23	24
	PD13	5,22	4,96	5,82	10	10	11	8,50	7,99	8,96	17	16	18	11,65	11,42	12,34	23	23	24
	PD21i	0,57	0,49	0,64	11	10	12	1,59	1,49	1,72	18	17	19	2,76	2,53	2,93	24	23	26
	PD22i	0,88	0,79	0,97	11	10	12	2,29	2,06	2,45	18	16	19	3,89	3,57	4,31	24	23	25
	PD23	10,18	9,34	10,80	11	10	11	15,99	14,99	17,35	17	16	18	22,00	21,15	23,13	24	23	24
	PGLs	0,40	0,34	0,44	225	204	245	1,41	1,29	1,52	803	746	875	2,67	2,48	2,91	1530	1424	1630

Tabuľka B.54: Vplyv dynamického rozsahu optimálneho riešenia: Porovnanie metód riešenia na 50-tich úlohách (1.2) s optimálnym riešením s dynamickým rozsahom $dr = 100000$.

Metóda	Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		# iterácií	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM		0,64	0,58	0,71	44	42	47	0,80	0,73	0,88	68	63	72	0,96	0,89	1,05	93	88	98
APG		0,14	0,13	0,15	98	93	103	0,86	0,82	0,90	608	581	638	4,07	3,80	4,37	2898	2696	3113
APGls		0,36	0,32	0,38	132	125	139	2,34	2,16	2,45	861	809	892	11,94	11,26	12,72	4361	4132	4641
DPG1		0,24	0,23	0,25	98	93	103	1,48	1,42	1,56	608	581	638	7,05	6,79	7,42	2916	2800	3073
DPG1ls		0,17	0,15	0,18	56	53	60	0,72	0,66	0,79	262	246	275	2,62	2,40	2,80	921	857	970
DPG2		0,24	0,23	0,25	98	93	103	1,47	1,17	1,56	603	482	638	2,61	1,65	4,25	1070	676	1750
DPG2ls		0,16	0,15	0,18	56	53	60	0,40	0,34	0,53	138	119	195	0,60	0,53	0,71	209	182	259
DPG3		0,24	0,23	0,25	98	93	103	1,47	1,24	1,55	604	508	638	2,74	2,12	4,32	1131	873	1788
DPG3ls		0,16	0,15	0,18	56	53	60	0,42	0,36	0,59	150	131	212	0,71	0,59	1,05	257	209	368
PD11i		0,23	0,20	0,27	12	10	14	0,65	0,56	0,74	18	16	21	1,16	1,08	1,25	25	23	27
PD11ir		0,25	0,22	0,30	12	10	14	0,30	0,26	0,34	16	14	18	0,33	0,29	0,37	20	18	22
PD12i		0,39	0,34	0,44	11	10	11	1,06	0,99	1,13	17	16	17	1,82	1,69	1,97	23	22	24
PD13		5,24	5,02	5,90	10	10	11	8,62	8,06	8,95	17	16	17	11,79	11,28	12,45	23	22	24
PD21i		0,57	0,48	0,64	11	10	13	1,58	1,47	1,79	18	17	19	2,78	2,60	3,01	24	23	26
PD22i		0,88	0,78	0,99	11	10	12	2,31	2,12	2,46	18	17	19	3,94	3,63	4,21	24	23	25
PD23		10,25	9,39	10,91	11	10	11	16,25	15,53	17,30	17	16	18	22,28	21,38	23,18	23	23	24
PGLs		0,40	0,37	0,44	224	209	241	1,41	1,30	1,51	805	746	872	2,68	2,49	2,83	1535	1423	1635

Tabuľka B.55: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 100$ a $n = 1000$.

Metóda	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	1,2E-3	9,4E-4	5,5E-3	26	21	32	2,4E-3	1,5E-3	6,8E-3	83	51	243	3,8E-3	2,3E-3	8,1E-3	147	89	303
APG	4,6E-4	2,2E-4	2,3E-3	39	24	56	1,7E-3	7,4E-4	4,0E-3	172	86	279	5,4E-3	2,5E-3	1,4E-2	597	291	1321
APGls	6,4E-4	2,7E-4	3,9E-3	36	22	53	2,2E-3	9,1E-4	6,1E-3	156	77	261	6,8E-3	3,1E-3	1,9E-2	539	257	1246
DPG1	6,3E-4	2,9E-4	3,3E-3	39	24	56	2,3E-3	1,0E-3	6,2E-3	173	89	336	7,8E-3	3,5E-3	1,8E-2	652	311	1294
DPG1ls	4,7E-4	2,3E-4	4,3E-3	23	16	32	1,4E-3	6,8E-4	6,2E-3	85	52	154	3,7E-3	1,5E-3	9,8E-3	262	116	549
DPG2	6,1E-4	2,9E-4	3,1E-3	39	24	56	1,7E-3	6,7E-4	4,9E-3	123	57	282	2,7E-3	1,2E-3	7,0E-3	202	93	347
DPG2ls	5,1E-4	2,3E-4	4,1E-3	23	16	32	1,2E-3	4,9E-4	5,7E-3	65	35	145	1,8E-3	8,6E-4	6,8E-3	105	58	182
DPG3	6,1E-4	2,9E-4	3,3E-3	39	24	56	1,8E-3	7,5E-4	4,5E-3	129	66	314	2,8E-3	1,3E-3	6,2E-3	217	112	399
DPG3ls	4,8E-4	2,2E-4	4,2E-3	23	16	32	1,2E-3	5,2E-4	5,1E-3	69	40	180	1,8E-3	8,2E-4	5,8E-3	112	63	226
PD11i	3,2E-3	1,9E-3	2,0E-2	10	9	11	5,9E-3	3,6E-3	2,3E-2	17	16	18	8,8E-3	5,6E-3	2,7E-2	23	22	24
PD11ir	3,6E-3	2,1E-3	2,3E-2	10	9	11	4,1E-3	2,5E-3	2,5E-2	12	11	14	4,6E-3	2,8E-3	2,7E-2	16	13	19
PD12i	1,7E-2	1,2E-2	3,2E-2	10	9	11	3,7E-2	2,3E-2	7,4E-2	17	16	18	5,9E-2	3,5E-2	1,0E-1	23	22	24
PD21i	6,7E-3	4,7E-3	2,1E-2	10	9	11	1,3E-2	8,8E-3	2,7E-2	16	15	17	2,0E-2	1,3E-2	3,6E-2	22	22	23
PD22i	2,2E-2	1,7E-2	3,3E-2	10	9	10	5,1E-2	3,3E-2	6,9E-2	16	15	17	8,2E-2	4,8E-2	1,1E-1	22	22	23
PG	1,7E-3	6,3E-4	4,5E-3	212	79	428	1,1E-2	3,1E-3	6,0E-2	1481	420	7991	2,3E-2	3,9E-3	8,4E-2	3048	526	11700
PGls	6,2E-4	2,5E-4	3,8E-3	51	25	92	2,7E-3	8,3E-4	1,3E-2	269	87	1454	5,2E-3	1,0E-3	1,6E-2	528	107	1772

Tabuľka B.56: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 500$ a $n = 5000$.

Metóda	Presn. ϵ			10^{-3}			10^{-6}			10^{-9}		
	čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	1,2E-2	1,0E-2	1,7E-2	26	22	39	2,8E-2	2,2E-2	6,6E-2	82	59	217
APG	4,0E-3	2,5E-3	5,4E-3	35	21	49	1,9E-2	1,1E-2	2,4E-2	172	93	226
APGls	6,0E-3	4,0E-3	8,9E-3	33	20	47	2,7E-2	1,6E-2	4,1E-2	160	88	217
DPG1	5,3E-3	3,3E-3	7,1E-3	35	21	49	2,5E-2	1,4E-2	3,3E-2	172	93	228
DPG1ls	4,0E-3	2,9E-3	5,4E-3	21	14	31	1,6E-2	1,0E-2	2,1E-2	87	48	112
DPG2	5,3E-3	3,3E-3	7,4E-3	35	21	49	2,1E-2	1,4E-2	3,2E-2	138	90	207
DPG2ls	4,0E-3	2,9E-3	6,2E-3	21	14	31	1,3E-2	6,9E-3	2,4E-2	71	34	120
DPG3	5,3E-3	3,3E-3	7,1E-3	35	21	49	2,1E-2	1,5E-2	3,6E-2	146	93	250
DPG3ls	4,0E-3	2,9E-3	6,1E-3	21	14	31	1,4E-2	7,5E-3	2,4E-2	75	36	137
PD11i	3,5E-2	2,8E-2	4,7E-2	12	11	12	7,9E-2	5,9E-2	1,1E-1	18	17	19
PD11lr	3,6E-2	2,8E-2	4,5E-2	12	11	12	4,1E-2	3,3E-2	5,5E-2	14	13	15
PD12i	9,2E-2	7,2E-2	1,3E-1	12	11	12	2,4E-1	1,8E-1	3,4E-1	18	17	19
PD21i	7,6E-2	6,0E-2	9,2E-2	11	10	11	1,8E-1	1,3E-1	2,7E-1	17	17	18
PD22i	1,6E-1	1,2E-1	2,0E-1	11	10	11	4,1E-1	3,1E-1	6,2E-1	17	17	18
PG	1,8E-2	7,3E-3	3,2E-2	179	71	320	1,5E-1	3,8E-2	5,1E-1	1487	370	5349
PGls	5,9E-3	2,8E-3	1,1E-2	44	21	90	3,6E-2	1,0E-2	1,1E-1	283	77	884

Tabuľka B.57: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 1000$ a $n = 10000$.

Metóda	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	3,3E-2	3,0E-2	4,2E-2	19	16	27	1,4E-1	6,0E-2	6,1E-1	123	45	612	3,5E-1	1,2E-1	1,5E+0	344	108	1445
APG	8,1E-3	6,0E-3	1,1E-2	35	25	47	3,8E-2	2,5E-2	5,9E-2	164	108	246	1,6E-1	9,4E-2	3,0E-1	687	400	1229
APGs	1,3E-2	8,9E-3	1,8E-2	33	23	46	5,6E-2	3,6E-2	8,5E-2	155	100	239	2,3E-1	1,3E-1	4,3E-1	641	373	1200
DPG1	1,1E-2	8,3E-3	1,6E-2	35	25	47	5,3E-2	3,5E-2	8,2E-2	164	108	246	2,3E-1	1,3E-1	4,2E-1	704	410	1275
DPG1ls	8,2E-3	6,0E-3	1,2E-2	21	15	29	3,3E-2	2,2E-2	5,0E-2	84	58	135	1,1E-1	7,1E-2	2,1E-1	305	194	583
DPG2	1,1E-2	8,6E-3	1,6E-2	35	25	47	5,0E-2	3,7E-2	6,9E-2	154	108	212	8,8E-2	5,6E-2	1,4E-1	268	181	414
DPG2ls	8,1E-3	5,4E-3	1,1E-2	21	15	29	2,8E-2	1,8E-2	4,6E-2	72	49	121	5,0E-2	2,9E-2	1,0E-1	130	79	244
DPG3	1,1E-2	8,4E-3	1,6E-2	35	25	47	5,1E-2	3,6E-2	7,1E-2	157	108	217	9,4E-2	6,7E-2	1,4E-1	291	204	414
DPG3ls	8,1E-3	5,6E-3	1,1E-2	21	15	29	2,9E-2	1,9E-2	5,1E-2	75	52	121	5,3E-2	3,3E-2	9,8E-2	139	90	264
PD11i	6,7E-2	5,5E-2	9,2E-2	12	11	13	1,6E-1	1,2E-1	2,2E-1	19	18	19	2,7E-1	1,9E-1	4,4E-1	25	24	26
PD11ir	7,1E-2	5,8E-2	9,3E-2	12	11	13	8,0E-2	6,7E-2	1,0E-1	14	13	15	8,2E-2	6,8E-2	1,1E-1	19	17	20
PD12i	1,7E-1	1,3E-1	2,3E-1	12	11	13	4,5E-1	3,3E-1	6,3E-1	19	18	20	8,1E-1	5,4E-1	1,2E+0	25	24	26
PD21i	1,2E-1	9,6E-2	1,6E-1	11	11	12	2,9E-1	2,2E-1	4,2E-1	18	17	18	4,8E-1	3,4E-1	7,9E-1	24	23	25
PD22i	2,2E-1	1,7E-1	3,1E-1	11	11	12	6,1E-1	4,5E-1	8,8E-1	18	17	19	1,1E+0	7,3E-1	1,7E+0	24	23	25
PG	3,6E-2	1,9E-2	6,3E-2	168	90	291	3,3E-1	1,2E-1	1,4E+0	1538	560	6585	9,4E-1	2,8E-1	3,4E+0	4438	1280	15601
PGls	1,2E-2	7,2E-3	2,0E-2	41	26	65	8,1E-2	3,0E-2	3,6E-1	292	105	1303	2,2E-1	6,2E-2	8,0E-1	796	203	2866

Tabuľka B.58: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 5000$ a $n = 50000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	8,1E-1	7,3E-1	1,3E+0	20	17	42	2,1E+0	1,4E+0	6,6E+0	82	51	308	6,2E+0	2,6E+0	2,0E+1	286	111	982
APG	2,2E-2	1,5E-2	2,8E-2	31	22	38	1,1E-1	7,5E-2	1,5E-1	149	108	216	4,5E-1	3,2E-1	6,0E-1	634	464	847
APGls	4,0E-2	2,8E-2	5,0E-2	30	21	37	1,9E-1	1,3E-1	2,7E-1	142	103	207	7,8E-1	5,7E-1	1,1E+0	599	446	815
DPG1	3,8E-2	2,7E-2	4,7E-2	31	22	38	1,8E-1	1,3E-1	2,6E-1	149	108	216	7,9E-1	5,7E-1	1,1E+0	644	465	877
DPG1ls	2,7E-2	2,0E-2	3,5E-2	19	14	24	1,1E-1	8,2E-2	1,7E-1	80	60	119	4,1E-1	2,7E-1	5,5E-1	294	191	397
DPG2	3,8E-2	2,6E-2	4,6E-2	31	22	38	1,8E-1	1,2E-1	2,6E-1	147	102	216	4,0E-1	2,5E-1	6,3E-1	328	199	504
DPG2ls	2,7E-2	1,9E-2	3,3E-2	19	14	24	9,9E-2	7,4E-2	1,9E-1	70	54	132	2,0E-1	1,3E-1	3,2E-1	138	93	230
DPG3	3,8E-2	2,6E-2	4,6E-2	31	22	38	1,8E-1	1,3E-1	2,6E-1	148	108	216	4,2E-1	2,7E-1	6,2E-1	340	224	513
DPG3ls	2,7E-2	1,9E-2	3,4E-2	19	14	24	1,0E-1	7,9E-2	2,2E-1	73	56	157	2,1E-1	1,4E-1	3,2E-1	149	101	232
PD11i	2,0E-1	1,6E-1	2,6E-1	14	13	15	4,8E-1	3,5E-1	6,4E-1	20	19	22	8,2E-1	6,2E-1	1,2E+0	27	26	28
PD11lr	2,1E-1	1,8E-1	2,7E-1	14	13	15	2,5E-1	2,1E-1	3,2E-1	16	15	18	2,7E-1	2,2E-1	3,4E-1	20	19	22
PD12i	4,5E-1	3,6E-1	5,2E-1	15	13	18	1,2E+0	9,4E-1	1,8E+0	21	19	24	2,1E+0	1,6E+0	3,2E+0	28	26	31
PD21i	5,0E-1	4,3E-1	6,0E-1	12	12	13	1,2E+0	9,7E-1	1,6E+0	19	18	20	2,0E+0	1,6E+0	3,2E+0	25	25	26
PD22i	8,5E-1	7,0E-1	1,0E+0	13	12	15	2,4E+0	1,9E+0	3,3E+0	20	18	21	4,3E+0	3,4E+0	6,7E+0	26	25	28
PG	9,1E-2	4,8E-2	1,3E-1	138	76	201	9,0E-1	4,7E-1	3,6E+0	1340	708	5378	3,6E+0	1,5E+0	1,2E+1	5374	2170	18235
PGls	3,0E-2	1,7E-2	4,0E-2	36	22	51	2,2E-1	1,1E-1	8,0E-1	263	137	981	8,3E-1	3,1E-1	2,8E+0	983	372	3316

Tabuľka B.59: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 10000$ a $n = 100000$.

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM	4,5E+0	4,0E+0	4,8E+0	28	24	33	8,3E+0	6,8E+0	9,3E+0	75	60	85	1,9E+1	1,2E+1	1,2E+2	205	119	1508
APG	3,8E-2	2,9E-2	4,9E-2	29	22	36	1,8E-1	1,3E-1	2,4E-1	136	98	184	7,5E-1	4,9E-1	1,1E+0	588	386	826
APGls	6,9E-2	5,1E-2	8,6E-2	28	21	35	3,2E-1	2,2E-1	4,3E-1	130	93	178	1,4E+0	9,1E-1	2,0E+0	562	368	802
DPG1	6,9E-2	5,4E-2	8,5E-2	29	22	36	3,2E-1	2,3E-1	4,3E-1	136	98	184	1,4E+0	9,0E-1	2,0E+0	593	386	851
DPG1ls	4,7E-2	3,7E-2	6,2E-2	18	15	23	2,0E-1	1,4E-1	2,6E-1	74	51	102	7,4E-1	4,5E-1	1,1E+0	279	166	427
DPG2	6,9E-2	5,3E-2	8,7E-2	29	22	36	3,2E-1	2,3E-1	4,3E-1	135	97	184	8,7E-1	5,3E-1	1,3E+0	372	225	571
DPG2ls	4,7E-2	3,8E-2	6,0E-2	18	15	23	1,7E-1	1,2E-1	2,3E-1	65	44	88	3,8E-1	2,5E-1	6,9E-1	140	89	257
DPG3	6,8E-2	5,2E-2	8,5E-2	29	22	36	3,2E-1	2,3E-1	4,2E-1	136	98	184	9,0E-1	5,0E-1	1,3E+0	388	211	586
DPG3ls	4,7E-2	3,8E-2	6,0E-2	18	15	23	1,8E-1	1,2E-1	2,6E-1	69	45	100	4,0E-1	2,6E-1	8,9E-1	154	101	341
PD11i	4,0E-1	3,4E-1	5,0E-1	15	13	16	9,4E-1	7,1E-1	1,4E+0	21	20	23	1,6E+0	1,2E+0	2,3E+0	27	26	30
PD11ir	4,2E-1	3,6E-1	5,2E-1	15	13	16	5,0E-1	4,2E-1	6,5E-1	17	15	19	5,4E-1	4,4E-1	6,8E-1	21	19	23
PD12i	9,2E-1	7,3E-1	1,1E+0	16	14	19	2,4E+0	1,9E+0	3,4E+0	23	20	25	4,4E+0	3,2E+0	6,0E+0	29	27	32
PD21i	1,1E+0	9,1E-1	1,5E+0	13	12	15	2,5E+0	2,0E+0	3,6E+0	20	19	21	4,3E+0	3,2E+0	5,9E+0	26	25	27
PD22i	2,1E+0	1,7E+0	2,5E+0	16	13	19	5,7E+0	4,6E+0	7,7E+0	22	20	26	1,0E+1	8,1E+0	1,4E+1	29	26	32
PG	1,5E-1	9,5E-2	2,2E-1	122	77	178	1,4E+0	6,5E-1	3,0E+0	1162	534	2448	6,4E+0	2,1E+0	2,5E+1	5281	1718	20745
PGls	4,9E-2	3,2E-2	6,8E-2	33	22	46	3,5E-1	1,5E-1	6,7E-1	230	104	448	1,5E+0	5,9E-1	5,9E+0	983	397	3796

Tabuľka B.60: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 50000$ a $n = 500000$.

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
ADMM																								
APG	2,2E-1	1,7E-1	2,5E-1	27	21	31	1,0E+0	7,8E-1	1,2E+0	128	96	152	4,5E+0	3,4E+0	5,9E+0	561	420	729						
APGls	3,7E-1	2,9E-1	4,3E-1	26	20	30	1,8E+0	1,3E+0	2,1E+0	123	91	145	7,7E+0	5,7E+0	1,0E+1	535	397	712						
DPG1	3,7E-1	2,9E-1	4,2E-1	27	21	31	1,7E+0	1,3E+0	2,1E+0	128	96	152	7,6E+0	5,7E+0	9,9E+0	559	421	730						
DPG1ls	2,5E-1	2,1E-1	2,9E-1	17	14	19	1,1E+0	8,8E-1	1,3E+0	71	58	85	4,2E+0	3,1E+0	5,6E+0	274	205	372						
DPG2	3,7E-1	2,9E-1	4,3E-1	27	21	31	1,7E+0	1,3E+0	2,1E+0	128	96	152	6,5E+0	3,5E+0	1,1E+1	476	254	832						
DPG2ls	2,5E-1	2,1E-1	2,9E-1	17	14	19	9,7E-1	6,6E-1	1,2E+0	63	43	80	2,7E+0	1,8E+0	6,2E+0	173	115	402						
DPG3	3,7E-1	2,9E-1	4,3E-1	27	21	31	1,7E+0	1,3E+0	2,1E+0	128	96	152	6,7E+0	3,5E+0	1,3E+1	490	260	970						
DPG3ls	2,5E-1	2,1E-1	2,9E-1	17	14	19	1,0E+0	7,0E-1	1,3E+0	66	46	83	2,9E+0	1,8E+0	6,9E+0	191	115	451						
PD11i	2,6E+0	2,0E+0	3,2E+0	18	15	21	6,2E+0	4,7E+0	8,4E+0	25	22	29	1,1E+1	7,3E+0	1,5E+1	32	28	35						
PD11ir	2,8E+0	2,2E+0	3,4E+0	18	15	21	3,3E+0	2,7E+0	3,9E+0	21	17	24	3,4E+0	2,8E+0	4,1E+0	25	22	28						
PD12i	6,2E+0	5,0E+0	7,1E+0	21	16	24	1,7E+1	1,3E+1	2,8E+1	30	23	34	3,3E+1	2,4E+1	5,0E+1	36	30	41						
PD21i	6,3E+0	5,4E+0	7,2E+0	17	15	22	1,6E+1	1,2E+1	2,0E+1	24	22	29	2,7E+1	2,0E+1	3,9E+1	31	28	35						
PD22i	1,2E+1			22			3,2E+1			31			5,8E+1			37								
PG	8,3E-1			115			8,1E+0			1129			4,2E+1			5873								
PGls	2,6E-1	1,8E-1	3,5E-1	30	21	39	2,0E+0	1,0E+0	3,3E+0	218	112	371	1,6E+1	5,5E+0	9,1E+1	1796	605	10091						

Tabuľka B.61: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.2) s rozmermi $m = 100000$ a $n = 1000000$.

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácii		čas		max		min		pr		max		min		pr		max		min		pr	
	pr	max	min	max	pr	max	min	max	pr	max	min	max	pr	max	min	max	pr	max	min	max	pr	max	min	max
APG	4,3E-1	3,4E-1	5,0E-1	5,0E-1	25	20	29	29	2,0E+0	1,6E+0	2,4E+0	2,4E+0	120	93	142	142	8,8E+0	6,0E+0	1,1E+1	1,1E+1	519	351	656	656
APGls	7,3E-1	5,7E-1	8,4E-1	8,4E-1	24	19	28	28	3,5E+0	2,6E+0	4,0E+0	4,0E+0	116	88	135	135	1,5E+1	9,9E+0	1,9E+1	1,9E+1	498	334	637	637
DPG1	7,3E-1	5,8E-1	8,4E-1	8,4E-1	25	20	29	29	3,4E+0	2,6E+0	4,0E+0	4,0E+0	120	93	142	142	1,5E+1	9,9E+0	1,9E+1	1,9E+1	519	351	656	656
DPG1ls	4,9E-1	4,0E-1	5,7E-1	5,7E-1	16	13	18	18	2,1E+0	1,6E+0	2,5E+0	2,5E+0	66	50	79	79	8,1E+0	5,5E+0	1,1E+1	1,1E+1	255	175	356	356
DPG2	7,2E-1	5,8E-1	8,3E-1	8,3E-1	25	20	29	29	3,4E+0	2,6E+0	4,0E+0	4,0E+0	120	93	142	142	1,3E+1	7,1E+0	1,8E+1	1,8E+1	474	250	656	656
DPG2ls	4,9E-1	4,0E-1	5,7E-1	5,7E-1	16	13	18	18	1,9E+0	1,4E+0	2,5E+0	2,5E+0	59	45	79	79	5,8E+0	3,5E+0	1,3E+1	1,3E+1	178	108	390	390
DPG3	7,3E-1	5,8E-1	8,4E-1	8,4E-1	25	20	29	29	3,4E+0	2,6E+0	4,0E+0	4,0E+0	120	93	142	142	1,4E+1	7,4E+0	1,8E+1	1,8E+1	480	259	656	656
DPG3ls	4,9E-1	4,0E-1	5,7E-1	5,7E-1	16	13	18	18	2,0E+0	1,4E+0	2,5E+0	2,5E+0	61	45	79	79	6,3E+0	3,9E+0	1,5E+1	1,5E+1	197	121	457	457
PD1i	5,4E+0	4,4E+0	6,3E+0	6,3E+0	20	17	24	24	1,3E+1	9,5E+0	1,8E+1	1,8E+1	29	24	32	32	2,2E+1	1,6E+1	3,1E+1	3,1E+1	35	30	39	39
PD11ir	5,8E+0	4,8E+0	6,8E+0	6,8E+0	20	17	24	24	6,9E+0	5,6E+0	9,5E+0	9,5E+0	23	19	27	27	7,1E+0	5,7E+0	9,7E+0	9,7E+0	28	24	32	32
PD12i	1,3E+1				26				3,5E+1				38				6,5E+1				45			
PD21i	1,7E+1				28				3,7E+1				37				5,8E+1				44			
PD22i	2,4E+1				27				5,8E+1				37				1,3E+2				46			
PG	1,4E+0				89				1,4E+1				909				1,9E+2				12203			
PGls	5,0E-1	3,9E-1	7,2E-1	7,2E-1	27	21	38	38	3,7E+0	2,2E+0	5,2E+0	5,2E+0	190	111	271	271	3,4E+1	1,1E+1	1,8E+2	1,8E+2	1733	544	9207	9207

Tabuľka B.62: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.7) s hodnotou regularizačného parametra $\lambda = 0.8\lambda_{max}$.

Presnosť ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácii			čas			# iterácii			čas			# iterácii		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	APG	0,00	0,00	0,03	2	1	20	0,17	0,10	0,31	120	70	221	1,24	0,11	1,64	877	77	1173
	APGls	0,01	0,00	0,02	1	1	6	0,12	0,07	0,25	40	21	86	0,65	0,18	0,98	233	60	345
	DPG1	0,01	0,00	0,05	2	1	20	0,36	0,18	0,58	144	70	232	2,64	0,20	3,18	1060	77	1291
	DPG1ls	0,01	0,00	0,03	1	1	6	0,10	0,06	0,17	33	19	54	0,45	0,13	0,58	151	43	196
	DPG2	0,01	0,00	0,05	2	1	20	0,23	0,18	0,29	89	70	115	0,39	0,20	0,51	151	77	200
	DPG2ls	0,01	0,01	0,03	1	1	6	0,07	0,06	0,09	22	19	28	0,11	0,09	0,17	34	26	51
	DPG3	0,01	0,00	0,05	2	1	20	0,24	0,18	0,36	95	70	138	0,49	0,20	0,63	194	77	248
	DPG3ls	0,01	0,00	0,03	1	1	6	0,08	0,06	0,12	24	19	37	0,16	0,11	0,21	50	38	67
	PD11i	0,20	0,14	0,25	11	10	12	0,38	0,23	0,49	18	16	19	0,49	0,30	0,63	25	23	27
	PD11lr	0,23	0,17	0,27	11	10	12	0,27	0,17	0,35	12	11	13	0,27	0,17	0,35	16	14	17
	PD12i	0,45	0,33	0,52	10	10	10	0,70	0,44	0,90	16	16	17	0,88	0,56	1,11	23	23	23
	PD13	5,70	5,42	5,93	10	10	10	9,14	8,77	9,42	16	16	16	13,13	12,67	13,54	23	23	23
	PD21i	0,51	0,38	0,62	11	10	12	0,88	0,58	1,15	18	17	20	1,14	0,79	1,45	25	23	29
	PD22i	1,01	0,68	1,16	10	10	10	1,64	0,97	2,14	17	17	17	1,95	1,18	2,50	23	23	24
	PD23	10,10	9,64	10,50	10	10	10	17,09	16,51	17,59	17	17	17	23,07	22,44	23,68	23	23	23
	PGls	0,01	0,00	0,02	1	1	7	0,15	0,11	0,21	77	52	110	0,32	0,21	0,41	167	114	215

Tabuľka B.64: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.7) s hodnotou regularizačného parametra $\lambda = 0.5\lambda_{max}$.

Metóda	Presnosť ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácií			čas			# iterácií			čas			# iterácií		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG		0,10	0,07	0,11	66	49	77	0,63	0,43	0,72	440	306	515	2,74	1,98	3,21	1937	1415	2301
APGls		0,08	0,05	0,10	25	14	32	0,41	0,20	0,52	142	70	182	1,47	0,65	1,79	522	231	643
DPG1		0,17	0,13	0,19	66	49	77	1,10	0,84	1,28	440	335	515	4,91	3,74	5,74	1960	1503	2315
DPG1ls		0,07	0,05	0,08	19	13	24	0,29	0,18	0,36	96	59	117	0,94	0,62	1,14	311	207	378
DPG2		0,17	0,13	0,19	66	49	77	0,79	0,30	1,28	312	118	515	1,15	0,50	2,84	455	198	1142
DPG2ls		0,07	0,05	0,08	19	13	24	0,17	0,10	0,25	52	29	82	0,25	0,14	0,36	78	42	118
DPG3		0,17	0,13	0,19	66	49	77	0,84	0,37	1,27	333	148	515	1,29	0,63	2,92	513	248	1173
DPG3ls		0,06	0,05	0,08	19	13	24	0,20	0,13	0,28	64	42	93	0,32	0,21	0,48	102	68	155
PD11i		0,29	0,23	0,32	11	11	12	0,75	0,47	0,93	17	17	19	1,00	0,61	1,33	24	24	26
PD11ir		0,31	0,26	0,34	11	11	12	0,40	0,33	0,45	14	13	15	0,41	0,33	0,46	18	17	19
PD12i		0,64	0,49	0,78	10	10	11	1,45	0,90	1,83	17	17	17	1,78	1,15	2,33	23	23	24
PD13		5,86	5,49	6,43	10	10	11	9,77	9,35	10,02	17	17	17	13,25	12,87	13,52	23	23	23
PD21i		0,72	0,59	0,90	11	11	12	1,85	1,06	2,23	18	17	19	2,36	1,47	3,11	25	24	27
PD22i		1,70	1,33	1,92	11	11	11	3,30	2,15	4,04	17	17	18	4,11	2,62	5,33	24	23	24
PD23		11,23	10,81	11,58	11	11	11	17,35	16,75	18,56	17	17	18	24,39	23,89	24,91	24	24	24
PGls		0,09	0,05	0,12	46	25	62	0,38	0,20	0,48	199	108	257	0,70	0,40	0,91	374	215	494

Tabuľka B.65: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.7) s hodnotou regularizačného parametra $\lambda = 0.35\lambda_{max}$.

Metóda	Presnosť ϵ			10^{-3}			10^{-6}			10^{-9}		
	čas		# iterácií	čas		# iterácií	čas		# iterácií	čas		# iterácií
	pr	min		pr	min		pr	min		pr	min	
APG	0,14	0,10	73	98	0,16	110	0,92	0,68	485	4,30	2,93	3553
APGls	0,15	0,09	31	52	0,19	67	0,88	0,48	169	3,62	1,66	1782
DPG1	0,25	0,18	73	98	0,28	110	1,63	1,20	485	7,64	5,22	3573
DPG1ls	0,11	0,08	23	34	0,13	42	0,50	0,32	104	1,67	1,04	701
DPG2	0,25	0,18	73	98	0,28	110	1,58	0,71	281	2,43	1,05	1613
DPG2ls	0,11	0,08	23	34	0,13	42	0,27	0,14	45	0,40	0,23	215
DPG3	0,25	0,18	73	98	0,28	110	1,59	0,77	308	2,53	1,15	1652
DPG3ls	0,11	0,08	23	34	0,13	42	0,31	0,19	60	0,50	0,30	232
PD11i	0,36	0,30	11	12	0,42	13	1,09	0,81	18	1,50	1,02	26
PD11ir	0,39	0,32	11	12	0,44	13	0,44	0,36	14	0,45	0,37	20
PD12i	0,82	0,65	11	11	1,00	12	2,06	1,55	17	2,62	1,86	25
PD13	6,36	6,06	11	11	6,97	12	10,31	10,09	18	13,78	13,50	24
PD21i	0,88	0,80	12	12	1,11	13	2,72	1,98	18	3,58	2,42	26
PD22i	2,19	1,80	12	12	2,54	12	4,83	3,55	18	6,04	4,23	25
PD23	11,81	11,06	12	11	12,71	12	18,32	17,89	18	24,50	23,91	25
PGls	0,21	0,11	56	109	0,27	144	0,77	0,43	228	1,42	0,81	992

Tabuľka B.66: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.7) s hodnotou regularizačného parametra $\lambda = 0.2\lambda_{max}$.

Metóda	Presnosť ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácií			čas			# iterácií			čas			# iterácií		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG		0,24	0,19	0,27	171	132	192	1,68	1,28	1,92	1197	913	1367	8,60	6,02	10,00	6144	4325	7132
APGls		0,41	0,26	0,65	146	92	231	2,80	1,64	4,70	1007	591	1687	14,02	7,47	25,08	5057	2682	9031
DPG1		0,43	0,33	0,48	171	132	192	2,97	2,27	3,39	1197	913	1367	15,36	11,06	17,82	6212	4462	7207
DPG1ls		0,19	0,15	0,22	63	48	71	1,00	0,71	1,12	336	238	377	3,63	2,48	4,22	1226	840	1424
DPG2		0,43	0,33	0,48	171	132	192	2,87	1,86	3,37	1155	749	1367	4,10	2,51	5,96	1650	1010	2398
DPG2ls		0,19	0,15	0,22	63	48	71	0,63	0,38	0,98	207	122	327	0,86	0,59	1,29	282	188	426
DPG3		0,43	0,33	0,48	171	132	192	2,88	1,98	3,40	1163	799	1367	4,48	3,08	6,11	1810	1239	2450
DPG3ls		0,19	0,15	0,22	63	48	71	0,68	0,44	0,98	226	147	327	1,01	0,70	1,47	338	232	491
PD11i		0,48	0,38	0,57	13	12	14	1,78	1,43	1,97	19	19	21	2,55	1,98	2,96	26	25	27
PD11ir		0,45	0,39	0,55	13	12	14	0,56	0,48	0,63	17	16	18	0,61	0,53	0,70	21	20	22
PD12i		1,14	0,89	1,29	13	12	13	3,13	2,62	3,51	19	19	20	4,25	3,44	4,88	25	25	26
PD13		7,29	6,79	7,69	13	12	13	10,88	10,55	11,31	19	19	20	14,55	14,11	15,18	25	25	26
PD21i		1,25	1,03	1,46	13	12	14	4,31	3,48	4,87	19	19	20	6,18	4,80	7,09	26	25	27
PD22i		3,08	2,41	3,60	12	12	13	7,71	6,18	8,67	19	18	19	10,04	7,97	11,48	25	25	26
PD23		12,42	11,85	13,45	12	12	13	19,15	17,99	19,69	19	18	19	25,34	24,83	26,56	25	25	26
PGls		0,56	0,37	0,69	306	198	374	2,03	1,32	2,55	1114	723	1407	3,86	2,45	4,85	2116	1346	2673

Tabuľka B.67: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.7) s hodnotou regularizačného parametra $\lambda = 0.1\lambda_{max}$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas		# iterácií		čas		čas		# iterácií		čas		čas		# iterácií		čas	
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	0,45	0,36	0,50	318	253	348	3,23	2,55	3,58	2281	1809	2532	18,66	13,90	21,05	13214	9915	14831
APGls	0,99	0,54	1,23	352	192	432	7,10	3,64	8,90	2542	1312	3159	42,11	19,21	55,01	15117	6898	19558
DPG1	0,80	0,63	0,89	318	253	348	5,69	4,46	6,30	2279	1809	2515	33,27	24,44	37,59	13350	9864	15031
DPG1ls	0,32	0,26	0,35	105	86	115	1,69	1,38	1,90	566	462	629	6,83	5,32	7,51	2294	1790	2499
DPG2	0,80	0,63	0,89	318	253	348	5,14	2,89	6,32	2056	1165	2511	7,13	4,35	10,97	2850	1753	4403
DPG2ls	0,32	0,25	0,35	105	86	115	1,03	0,71	1,58	340	234	524	1,43	1,08	1,98	472	351	654
DPG3	0,80	0,63	0,87	318	253	348	5,20	3,04	6,29	2084	1215	2511	7,99	5,12	11,01	3203	2060	4435
DPG3ls	0,32	0,26	0,35	105	86	115	1,08	0,74	1,64	360	244	541	1,68	1,22	2,35	561	404	781
PD11i	0,62	0,52	0,78	14	14	15	3,02	2,37	3,33	21	20	21	4,52	3,59	5,10	27	26	28
PD11ir	0,53	0,43	0,62	14	13	15	0,84	0,68	0,96	18	18	19	0,96	0,77	1,09	22	22	23
PD12i	1,49	1,23	1,81	14	14	15	5,14	4,24	5,59	21	20	21	7,16	5,66	7,98	27	26	27
PD13	8,02	7,39	8,68	14	13	15	11,50	11,19	12,26	20	20	21	15,36	14,70	15,86	27	26	27
PD21i	1,58	1,31	2,04	14	13	15	7,36	5,68	8,08	21	20	21	10,77	8,50	12,11	27	26	28
PD22i	4,89	3,56	5,70	14	13	15	12,50	10,20	13,88	20	20	21	17,05	13,40	18,68	27	26	27
PD23	14,11	13,08	14,83	14	13	14	20,34	19,86	21,40	20	20	21	26,89	26,05	28,16	27	26	27
PGls	1,31	1,02	1,51	705	546	811	4,85	3,43	5,66	2627	1861	3047	9,50	6,55	10,90	5146	3558	5896

Tabuľka B.69: Vplyv regularizačného parametra λ : Porovnanie metód riešenia na 50-tich úlohách (1.7) s hodnotou regularizačného parametra $\lambda = 0.01\lambda_{max}$.

Metóda	Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácií			čas			# iterácií			čas			# iterácií		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	2,16				1557			15,66			11156			112,17			80497		
APGls	3,69				1335			26,12			9579			186,97			68766		
DPG1	3,81				1557			27,34			11195			199,87			81885		
DPG1ls	0,62	0,58	0,58	0,67	215	205	225	3,39	3,19	3,58	1149	1077	1219	16,06	14,71	17,10	5406	4957	5738
DPG2	4,17	3,47	3,47	4,58	1669	1398	1812	12,73	10,48	14,27	5101	4217	5749	21,43	16,74	24,25	8590	6749	9648
DPG2ls	0,64	0,58	0,58	0,68	215	205	225	1,86	1,46	2,53	621	484	844	2,75	2,27	3,43	918	753	1141
DPG3	4,21	3,49	3,49	4,62	1687	1398	1860	14,51	10,78	19,15	5824	4313	7699	30,15	19,73	51,44	12110	7921	20694
DPG3ls	0,64	0,57	0,57	0,68	215	205	225	1,89	1,51	2,61	633	506	873	3,47	2,63	4,88	1166	880	1650
PD11i	0,98	0,67	0,67	1,21	17	16	18	10,83	9,10	12,19	23	23	25	18,30	14,14	20,31	30	29	31
PD11ir	1,07	0,77	0,77	1,28	17	16	18	4,22	3,28	4,85	22	21	23	5,08	3,95	5,69	26	25	27
PD12i	2,42	1,82	1,82	3,14	17	16	17	16,46	13,95	20,77	23	22	24	25,07	20,05	29,70	29	29	30
PD13	9,27	8,89	8,89	9,87	16	16	17	13,08	12,25	13,50	23	22	23	16,71	16,16	17,61	29	29	30
PD21i	2,08				17			26,46			24			40,63			30		
PD22i	7,75				16			40,54			23			56,62			29		
PD23	16,13	15,11	15,11	16,92	16	15	16	22,62	21,90	24,06	22	22	23	29,03	27,91	30,30	29	28	29
PGLs	4,30	3,91	3,91	4,72	2327	2114	2533	17,37	15,16	20,68	9411	8185	11109	36,86	31,38	45,07	19969	17047	24209

Tabuľka B.71: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 1200$ a $n = 12000$.

Presn. ϵ	10^{-3}										10^{-6}										10^{-9}									
	čas					# iterácií					čas					# iterácií					čas					# iterácií				
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																														
APG	0,25	0,21	0,28	359	311	403				1,75	1,55	1,93	2525	2248	2827				10,50	8,70	12,10	14945	12516	17136						
APGls	0,49	0,29	0,68	357	209	494				3,43	1,89	4,89	2508	1391	3556				20,48	10,02	30,85	14985	7372	22501						
DPG1	0,43	0,37	0,48	359	311	403				3,04	2,63	3,40	2527	2190	2833				18,33	15,31	20,97	15176	12760	17469						
DPG1ls	0,15	0,13	0,16	97	87	106				0,74	0,67	0,82	507	453	554				3,09	2,74	3,50	2122	1876	2387						
DPG2	0,43	0,37	0,48	359	311	403				2,15	1,27	3,05	1784	1055	2520				3,01	2,09	4,20	2494	1745	3451						
DPG2ls	0,15	0,13	0,16	97	87	106				0,44	0,32	0,60	295	220	402				0,64	0,51	0,78	431	341	525						
DPG3	0,43	0,37	0,48	359	311	403				2,16	1,22	3,01	1796	1017	2520				3,44	2,18	4,52	2861	1820	3778						
DPG3ls	0,15	0,12	0,16	97	87	106				0,45	0,35	0,62	307	236	419				0,77	0,61	1,13	522	417	776						
PD1i	0,38	0,30	0,48	16	15	16				1,41	1,23	1,54	22	21	23				2,00	1,70	2,27	28	28	29						
PD1lr	0,30	0,26	0,35	15	14	16				0,42	0,36	0,45	19	18	20				0,46	0,40	0,50	24	23	24						
PD12i	0,86	0,72	1,03	15	15	16				2,46	2,11	2,71	22	21	23				3,29	2,85	3,60	28	27	29						
PD13	2,50	2,30	2,67	15	14	16				3,57	3,47	3,68	22	21	22				4,64	4,47	4,83	28	27	29						
PD21i	1,14	0,90	1,36	16	15	16				4,00	3,48	4,56	22	21	23				5,59	4,70	6,22	29	28	31						
PD22i	2,74	2,24	3,21	15	14	15				6,41	5,46	7,12	21	20	22				8,36	7,18	9,19	28	27	28						
PD23	4,35	4,03	4,63	15	14	16				6,20	6,02	6,41	21	21	22				8,08	7,76	8,40	28	27	29						
PGls	0,47	0,40	0,53	604	521	689				1,95	1,61	2,31	2526	2072	2971				3,97	3,23	4,86	5146	4206	6364						

Tabuľka B.73: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 4800$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		# iterácií	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG		0,91	0,72	1,02	276	217	310				6,65	5,06	7,51	2024	1537	2284	36,98	27,50	43,12
APGls		2,09			358						15,28			2644			87,07		15106
DPG1		1,67			290						11,97			2099			66,67		11677
DPG1ls		0,69	0,56	0,78	112	90	123				3,89	3,12	4,34	632	500	703	15,41	11,66	17,74
DPG2		1,58	1,26	1,78	276	217	310				11,46	8,87	13,10	2000	1537	2284	17,16	12,22	25,19
DPG2ls		0,70	0,57	0,76	112	90	123				2,37	1,43	3,41	381	228	552	3,16	2,20	4,12
DPG3		1,58	1,24	1,77	276	217	310				11,46	8,81	13,06	2005	1537	2284	18,48	13,09	25,23
DPG3ls		0,69	0,55	0,77	112	90	123				2,52	1,54	3,53	410	250	577	3,79	2,73	5,14
PD11i		1,00	0,73	1,19	13	12	14				5,67	4,60	6,56	19	19	20	9,12	7,08	10,16
PD11ir		0,95	0,71	1,11	13	12	13				1,93	1,49	2,13	17	17	18	2,36	1,78	2,73
PD12i		2,62	2,11	3,00	13	12	13				9,96	8,15	10,76	19	19	20	15,24	12,25	16,76
PD13		25,39			13						37,67			19			52,09		26
PD21i		2,41	1,87	2,78	13	12	13				12,49	10,04	14,36	19	19	20	20,18	15,92	22,73
PD22i		7,86	6,21	9,51	13	12	13				23,74	19,60	26,14	19	18	19	33,56	26,74	37,53
PD23		41,22			12						64,81			19			85,61		25
PGLs		3,19	2,30	3,76	820	588	962				10,90	7,96	13,48	2797	2039	3457	20,57	14,66	25,65
																	5277	3753	6575

Tabuľka B.74: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 9600$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácií		čas		# iterácií		čas		# iterácií		čas		# iterácií		# iterácií	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	APG	1,78	1,58	1,98	237	211	265	13,23	11,53	14,80	1755	1539	1979	69,52	59,06	80,28	9232	7894	10736
	DPG1ls	1,61	1,36	1,77	116	98	129	9,35	7,82	10,19	683	576	751	36,12	30,14	41,47	2646	2227	3065
	DPG2	3,04	2,69	3,41	237	211	265	22,36	15,93	25,39	1744	1242	1979	40,34	19,94	59,34	3144	1553	4634
	DPG2ls	1,61	1,35	1,80	116	98	129	5,75	3,42	8,39	415	243	609	7,45	5,27	9,87	536	376	712
	DPG3	3,03	2,68	3,36	237	211	265	22,35	16,66	25,10	1746	1300	1979	41,45	21,49	60,28	3236	1677	4709
	DPG3ls	1,61	1,36	1,81	116	98	129	6,15	3,82	8,67	449	278	639	8,51	6,16	11,33	622	453	820
	PD11i	1,46	1,05	1,68	12	11	12	9,61	8,55	11,20	18	18	19	17,34	15,30	18,96	25	24	25
	PD11ir	1,67	1,30	1,93	12	11	12	4,48	3,72	5,02	17	16	17	6,37	5,49	7,48	21	21	21
	PD12i	4,33	3,94	4,66	12	12	12	18,43	17,14	20,39	18	18	19	31,22	28,67	33,74	25	24	25
	PD21i	3,38	2,59	4,04	12	11	12	21,71	19,81	26,50	18	18	19	37,67	34,02	41,02	25	24	26
	PD22i	12,49	10,86	14,80	11	11	12	45,14	40,91	49,23	18	17	18	68,02	61,46	74,08	24	24	24
	PGLs	7,66	6,05	9,02	902	715	1052	24,37	19,83	28,87	2875	2353	3376	44,33	36,13	53,56	5232	4289	6280

Tabuľka B.75: Vplyv rozmeru m : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 12000$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácii			čas			# iterácii			čas			# iterácii		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	APG	2,23	2,02	2,44	225	204	247	16,57	15,13	18,19	1675	1520	1844	84,18	74,25	96,39	8510	7520	9764
	DPG1ls	2,15	1,93	2,37	118	105	129	12,53	11,37	13,95	697	632	772	47,48	42,03	53,25	2650	2343	2965
	DPG2	3,80	3,46	4,19	225	204	247	28,23	25,71	31,15	1675	1520	1844	52,65	33,46	85,43	3121	1983	5054
	DPG2ls	2,16	1,93	2,36	118	105	129	7,21	4,01	12,69	396	217	704	9,40	6,08	15,02	516	329	827
	DPG3	3,79	3,46	4,18	225	204	247	28,20	25,69	31,12	1675	1520	1844	53,72	37,35	86,61	3190	2215	5130
	DPG3ls	2,15	1,94	2,35	118	105	129	7,71	4,96	12,66	428	274	704	10,90	7,86	16,28	606	436	906
	PD1li	1,55	1,21	1,92	12	11	12	11,59	10,31	12,97	18	18	19	21,01	18,39	23,49	24	24	25
	PD1lir	1,87	1,57	2,29	12	11	12	5,79	4,98	6,49	17	16	17	8,56	7,35	9,50	21	20	21
	PD12i	5,46	4,18	5,77	12	11	12	23,01	21,73	25,14	18	18	18	37,98	34,29	40,49	24	24	25
	PD21i	3,69	3,06	4,61	11	11	12	25,74	22,20	29,05	18	17	19	45,46	40,95	49,41	24	24	25
	PD22i	16,39			11			51,75			17			83,28			24		
	PGls	10,15	8,71	11,55	916	784	1045	31,59	26,42	36,73	2858	2387	3312	57,29	46,71	67,63	5188	4224	6101

Tabuľka B.76: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 2400$ a $n = 3000$.

Presn. ϵ	10^{-3}										10^{-6}										10^{-9}									
	čas					# iterácii					čas					# iterácii					čas					# iterácii				
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																														
APG	0,05	0,04	0,05	144	128	164					0,32	0,28	0,37	997	872	1142					1,50	1,26	1,86	4690	3941	5853				
APGls	0,10	0,07	0,12	160	109	190					0,68	0,45	0,83	1137	745	1391					3,27	1,93	4,06	5501	3232	6858				
DPG1	0,08	0,07	0,10	144	128	164					0,57	0,46	0,66	997	872	1137					2,70	2,09	3,38	4746	4000	5840				
DPG1ls	0,05	0,04	0,05	75	66	83					0,27	0,22	0,30	423	377	480					0,97	0,79	1,18	1554	1307	1849				
DPG2	0,08	0,07	0,10	144	128	164					0,53	0,36	0,64	926	631	1104					0,71	0,47	1,13	1245	823	1933				
DPG2ls	0,05	0,04	0,05	75	66	83					0,15	0,10	0,27	241	144	425					0,21	0,15	0,32	328	224	507				
DPG3	0,08	0,07	0,10	144	128	164					0,54	0,38	0,64	936	660	1104					0,78	0,52	1,14	1360	904	1950				
DPG3ls	0,05	0,04	0,05	75	66	83					0,17	0,12	0,28	264	176	434					0,24	0,19	0,36	385	286	562				
PD11i	0,06	0,05	0,07	12	11	12					0,37	0,32	0,42	18	18	19					0,58	0,51	0,68	24	24	25				
PD11ir	0,07	0,05	0,08	12	11	12					0,12	0,10	0,13	16	16	16					0,14	0,13	0,16	20	20	21				
PD12i	0,16	0,13	0,20	12	12	13					0,73	0,65	0,84	19	18	19					1,05	0,94	1,21	25	25	26				
PD13	2,95	2,79	3,16	12	12	13					4,50	4,21	4,63	19	18	19					6,00	5,86	6,25	25	25	26				
PD21i	0,13	0,11	0,19	11	11	12					0,84	0,72	0,95	18	17	18					1,24	1,06	1,44	24	24	24				
PD22i	0,45	0,41	0,56	11	11	12					1,70	1,50	1,95	18	17	18					2,31	2,07	2,57	24	24	24				
PD23	3,95	3,80	4,28	11	11	12					6,31	5,92	6,58	18	17	18					8,51	8,32	8,69	24	24	24				
PGLs	0,16	0,14	0,20	432	358	516					0,54	0,45	0,70	1451	1210	1874					1,00	0,79	1,32	2698	2123	3512				

Tabuľka B.77: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 2400$ a $n = 6000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																		
APG	0,14	0,11	0,16	214	172	247	0,95	0,75	1,12	1507	1184	1777	5,03	3,74	6,16	7994	5983	9626
APGls	0,32	0,22	0,38	243	163	292	2,31	1,49	2,88	1759	1117	2190	12,57	7,73	15,90	9559	5842	12122
DPG1	0,24	0,19	0,27	214	172	247	1,67	1,31	1,96	1507	1184	1773	8,92	6,89	10,60	8076	6235	9627
DPG1ls	0,13	0,10	0,14	90	76	98	0,70	0,57	0,78	503	424	555	2,74	2,26	3,14	1964	1601	2239
DPG2	0,24	0,19	0,28	214	172	247	1,56	1,05	1,92	1404	941	1740	2,08	1,37	2,95	1872	1232	2657
DPG2ls	0,13	0,11	0,15	90	76	98	0,43	0,31	0,56	305	212	401	0,58	0,45	0,72	410	310	500
DPG3	0,24	0,19	0,27	214	172	247	1,57	1,09	1,90	1420	980	1740	2,31	1,48	3,09	2092	1338	2790
DPG3ls	0,13	0,11	0,14	90	76	98	0,46	0,33	0,59	328	234	417	0,67	0,51	0,91	480	352	645
PD1i	0,18	0,13	0,24	12	12	13	1,02	0,83	1,17	19	18	19	1,54	1,26	1,85	25	25	26
PD1lr	0,16	0,13	0,21	12	12	13	0,33	0,25	0,38	17	16	17	0,39	0,32	0,47	21	20	22
PD12i	0,47	0,41	0,53	13	13	13	1,84	1,50	2,10	19	19	20	2,69	2,20	3,07	26	25	26
PD13	4,55	4,10	4,69	13	12	13	6,73	6,56	7,10	19	19	20	9,06	8,71	9,31	26	25	26
PD21i	0,54	0,42	0,68	12	12	13	2,90	2,34	3,27	19	18	19	4,27	3,32	4,97	25	25	26
PD22i	1,60	1,37	1,95	12	12	13	5,08	4,22	5,67	19	18	19	6,94	5,54	7,97	25	25	26
PD23	6,93	6,75	7,12	12	12	12	10,59	10,17	11,15	18	18	19	14,39	14,10	14,82	25	24	25
PGls	0,42	0,31	0,50	576	430	691	1,46	1,09	1,82	2048	1512	2522	2,81	2,02	3,60	3932	2845	4982

Tabuľka B.79: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 2400$ a $n = 24000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	1,55	1,34	1,77	476	412	544	11,17	9,33	13,09	3445	2886	4036	69,46	56,11	79,86	21452	17436	24651
APGls	3,28			577			23,52			4169			155,79			27611		
DPG1	2,47			463			17,70			3303			113,73			21243		
DPG1ls	0,73	0,67	0,79	119	109	129	3,77	3,37	4,13	622	561	680	15,68	13,49	17,33	2592	2243	2855
DPG2	2,58	2,23	2,93	476	412	544	15,89	8,26	21,08	2934	1525	3883	21,82	12,74	30,00	4027	2353	5577
DPG2ls	0,73	0,67	0,79	119	109	129	2,35	1,68	3,41	385	274	565	3,31	2,64	4,47	541	431	739
DPG3	2,58	2,23	2,97	476	412	544	15,48	7,21	21,19	2859	1322	3883	24,43	14,28	38,54	4512	2657	7157
DPG3ls	0,73	0,67	0,79	119	109	129	2,43	1,72	3,50	401	282	583	3,98	3,01	5,28	656	496	879
PD1i	1,86	1,50	2,26	16	16	18	7,56	6,57	8,52	23	22	24	11,11	9,58	12,45	29	28	31
PD1ir	1,59	1,31	1,90	16	15	17	2,19	1,82	2,57	20	19	21	2,37	2,01	2,73	24	23	26
PD12i	4,33	3,64	5,04	16	15	16	12,26	11,18	13,23	22	22	23	17,07	15,20	18,57	29	28	29
PD13	16,05	14,99	16,62	16	15	16	22,37	21,75	23,32	22	22	23	29,01	27,96	29,97	29	28	29
PD21i	4,52	3,60	5,58	17	16	18	17,01	15,69	18,73	23	22	24	24,81	21,78	27,48	30	28	32
PD22i	12,42	10,71	13,73	16	15	16	28,27	24,74	31,55	22	22	23	37,95	33,41	41,88	29	28	29
PD23	30,53	29,77	32,44	16	16	17	42,69	40,82	44,42	23	22	23	54,57	53,78	57,62	29	29	30
PGls	3,22	2,74	3,81	829	707	985	12,24	10,01	14,04	3165	2590	3614	24,66	19,52	28,73	6378	5052	7437

Tabuľka B.80: Vplyv rozmeru n : Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 2400$ a $n = 48000$.

Presn. ϵ	10^{-3}										10^{-6}										10^{-9}									
	čas					# iterácií					čas					# iterácií					čas					# iterácií				
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	4,95			726									35,77			5271								243,94			35872			
DPG1ls	1,60	1,47	1,73	131	120	141							8,05	7,31	8,69	661	602	712						34,05	29,95	38,45	2797	2468	3151	
DPG2	7,77	6,89	8,69	692	616	772							45,10	20,84	58,89	4018	1861	5258						65,07	35,32	105,68	5794	3159	9373	
DPG2ls	1,60	1,46	1,73	131	120	141							4,93	3,42	6,94	403	279	567						7,06	5,58	9,03	575	453	736	
DPG3	7,78	6,92	8,71	692	616	772							36,54	21,67	59,13	3259	1925	5258						71,42	41,34	109,77	6371	3706	9739	
DPG3ls	1,60	1,46	1,73	131	120	141							5,07	3,53	7,11	416	289	584						8,76	6,21	10,92	720	510	894	
PD11i	4,99	4,08	5,82	22	20	24							16,94	14,80	19,32	28	26	31						24,23	20,79	29,03	34	33	37	
PD11ir	3,55	3,10	4,04	21	19	23							4,39	3,94	4,93	25	23	27						4,61	4,11	5,13	29	27	31	
PD12i	11,20	9,68	12,25	19	18	19							27,25	24,18	30,43	25	24	26						36,42	32,28	41,93	31	31	32	
PD13	35,48	33,25	37,13	19	18	20							47,35	44,40	48,88	25	24	26						59,48	57,44	60,24	32	31	32	
PD21i	11,69	9,19	13,77	21	19	26							38,22	34,14	44,54	28	26	33						53,67	45,58	64,02	34	32	39	
PD22i	32,15			19									65,28			25								84,25			32			
PD23	63,94			19									87,54			26								107,74			32			
PGLs	7,12	6,18	7,95	918	798	1027							27,43	22,87	33,43	3541	2954	4311						56,26	44,23	71,90	7263	5715	9267	

Tabuľka B.82: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 1200$ a $n = 6000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																		
APG	0,09	0,08	0,10	244	212	279	0,61	0,53	0,70	1703	1447	1932	3,33	2,80	3,91	9269	7661	10765
APGls	0,16	0,08	0,21	252	126	327	1,13	0,51	1,49	1765	806	2355	6,18	2,54	8,38	9668	3935	13136
DPG1	0,15	0,13	0,18	244	212	279	1,06	0,91	1,20	1704	1452	1933	5,84	4,92	6,96	9406	7805	11137
DPG1ls	0,06	0,05	0,07	86	74	93	0,32	0,28	0,35	463	404	503	1,28	1,13	1,46	1880	1637	2131
DPG2	0,15	0,13	0,18	244	212	279	0,82	0,50	1,15	1322	805	1854	1,11	0,74	1,52	1776	1183	2452
DPG2ls	0,06	0,05	0,07	86	74	93	0,18	0,13	0,26	263	190	386	0,27	0,21	0,35	383	301	491
DPG3	0,15	0,13	0,18	244	212	279	0,83	0,47	1,15	1344	751	1854	1,28	0,87	1,67	2058	1398	2705
DPG3ls	0,06	0,05	0,07	86	74	93	0,19	0,14	0,27	279	201	399	0,32	0,25	0,43	465	357	620
PD11i	0,15	0,11	0,18	14	13	14	0,64	0,56	0,73	20	19	20	0,94	0,79	1,06	26	26	27
PD11ir	0,13	0,11	0,17	13	13	14	0,18	0,15	0,21	17	17	18	0,21	0,17	0,24	22	21	22
PD12i	0,35	0,29	0,41	14	13	15	1,13	0,97	1,25	20	20	21	1,57	1,37	1,79	27	26	27
PD13	1,65	1,50	1,81	14	13	15	2,38	2,27	2,54	20	20	21	3,18	3,00	3,36	27	26	27
PD21i	0,37	0,27	0,43	14	13	14	1,40	1,21	1,59	20	19	21	2,01	1,64	2,29	26	25	27
PD22i	0,96	0,81	1,19	13	13	14	2,58	2,27	2,82	20	19	21	3,37	2,87	3,70	26	26	27
PD23	2,64	2,51	2,83	13	13	14	3,92	3,65	4,27	19	19	20	5,24	4,99	5,56	26	25	26
PGls	0,22	0,18	0,25	524	421	605	0,85	0,69	1,01	2074	1681	2440	1,69	1,37	2,05	4146	3318	4999

Tabuľka B.83: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 2400$ a $n = 12000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas		# iterácii		čas		# iterácii		čas		# iterácii		čas		# iterácii		# iterácii	
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	APG	0,45	0,36	0,50	318	253	348	3,23	2,55	3,58	2281	1809	2532	18,66	13,90	21,05	13214	9915	14831
	APGls	0,99	0,54	1,23	352	192	432	7,10	3,64	8,90	2542	1312	3159	42,11	19,21	55,01	15117	6898	19558
	DPG1	0,80	0,63	0,89	318	253	348	5,69	4,46	6,30	2279	1809	2515	33,27	24,44	37,59	13350	9864	15031
	DPG1ls	0,32	0,26	0,35	105	86	115	1,69	1,38	1,90	566	462	629	6,83	5,32	7,51	2294	1790	2499
	DPG2	0,80	0,63	0,89	318	253	348	5,14	2,89	6,32	2056	1165	2511	7,13	4,35	10,97	2850	1753	4403
	DPG2ls	0,32	0,25	0,35	105	86	115	1,03	0,71	1,58	340	234	524	1,43	1,08	1,98	472	351	654
	DPG3	0,80	0,63	0,87	318	253	348	5,20	3,04	6,29	2084	1215	2511	7,99	5,12	11,01	3203	2060	4435
	DPG3ls	0,32	0,26	0,35	105	86	115	1,08	0,74	1,64	360	244	541	1,68	1,22	2,35	561	404	781
	PD1li	0,62	0,52	0,78	14	14	15	3,02	2,37	3,33	21	20	21	4,52	3,59	5,10	27	26	28
	PD1lir	0,53	0,43	0,62	14	13	15	0,84	0,68	0,96	18	18	19	0,96	0,77	1,09	22	22	23
	PD12i	1,49	1,23	1,81	14	14	15	5,14	4,24	5,59	21	20	21	7,16	5,66	7,98	27	26	27
	PD13	8,02	7,39	8,68	14	13	15	11,50	11,19	12,26	20	20	21	15,36	14,70	15,86	27	26	27
	PD21i	1,58	1,31	2,04	14	13	15	7,36	5,68	8,08	21	20	21	10,77	8,50	12,11	27	26	28
	PD22i	4,89	3,56	5,70	14	13	15	12,50	10,20	13,88	20	20	21	17,05	13,40	18,68	27	26	27
	PD23	14,11	13,08	14,83	14	13	14	20,34	19,86	21,40	20	20	21	26,89	26,05	28,16	27	26	27
	PGls	1,31	1,02	1,51	705	546	811	4,85	3,43	5,66	2627	1861	3047	9,50	6,55	10,90	5146	3558	5896

Tabuľka B.84: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 4800$ a $n = 24000$.

Presn. ϵ	10^{-3}										10^{-6}										10^{-9}									
	čas					# iterácii					čas					# iterácii					čas					# iterácii				
	metoda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max		
APG	2,94	2,41	3,29	424	346	472					21,72	17,35	24,25	3133	2503	3491					132,27	102,90	150,32	19084	14855	21640				
APGls	6,38			535							45,89			3971							298,74			25937						
DPG1	4,94			429							36,33			3158							227,39			19779						
DPG1ls	1,67	1,35	1,88	134	108	151					9,00	7,22	9,96	731	588	810					36,99	29,40	42,65	3011	2396	3464				
DPG2	4,92	4,02	5,49	424	346	472					35,80	25,02	40,56	3088	2156	3491					55,81	32,54	84,49	4814	2802	7312				
DPG2ls	1,67	1,35	1,87	134	108	151					5,54	3,91	8,07	448	312	654					7,44	5,72	9,83	598	457	794				
DPG3	4,92	4,01	5,47	424	346	472					35,84	25,64	40,45	3093	2213	3491					59,34	39,43	84,81	5122	3405	7352				
DPG3ls	1,67	1,34	1,87	134	108	151					5,82	4,12	8,25	472	333	671					8,99	6,68	12,08	730	541	979				
PD11i	2,86	2,31	3,44	15	14	16					14,08	11,76	15,64	21	21	22					22,07	18,94	24,80	27	27	28				
PD11ir	2,72	2,32	3,17	14	14	15					4,82	3,92	5,40	19	18	20					5,49	4,43	6,16	23	23	24				
PD12i	7,04	5,83	8,57	14	14	15					24,34	21,29	26,95	21	20	21					35,29	30,32	38,92	27	26	28				
PD13	49,14	47,26	53,39	14	14	15					71,77	68,36	74,89	21	20	21					94,30	91,81	98,94	27	27	28				
PD21i	7,00	5,39	8,08	15	14	16					30,80	26,03	34,01	21	21	22					48,39	40,89	54,77	28	27	29				
PD22i	21,88	19,28	24,37	14	14	15					54,89	46,69	61,26	20	20	21					78,13	65,44	87,09	27	26	28				
PD23	88,30			14							125,54			20							169,58			27						
PGls	8,22	6,26	9,57	1051	799	1224					28,65	21,30	34,60	3671	2728	4424					55,26	40,05	67,70	7082	5135	8654				

Tabuľka B.85: Vplyv rozmerov $m \times n$: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 9600$ a $n = 48000$.

Presn. ϵ	Metóda	10^{-3}						10^{-6}						10^{-9}					
		čas			# iterácií			čas			# iterácií			čas			# iterácií		
		pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
	APG	15,73			519			118,80			3922			752,11			24833		
	DPG1s	9,96	8,51	10,99	182	156	198	55,44	46,14	61,54	1020	855	1111	234,21	189,30	263,80	4312	3516	4789
	DPG2	26,09			519			197,46			3922			312,00			6180		
	DPG2ls	9,95	8,56	10,87	182	156	198	31,79	18,22	47,94	582	334	877	42,37	27,48	58,58	774	504	1068
	DPG3	26,57			519			200,70			3922			333,63			6520		
	DPG3ls	9,96	8,54	10,86	182	156	198	33,57	20,57	50,08	617	383	905	52,28	38,39	70,86	961	709	1281
	PD1i	12,92	10,50	15,17	16	16	18	62,52	51,77	72,33	23	22	25	106,80	91,92	119,36	29	28	31
	PD1lr	13,54	11,09	15,77	16	15	17	25,55	21,31	29,86	21	20	22	30,74	26,03	35,00	25	24	26
	PD12i	37,06	30,06	40,60	15	14	15	112,94	99,32	122,91	21	21	22	173,49	148,03	191,14	27	27	28
	PD13	357,58			15			498,90			21			643,65			27		
	PD2li	32,03			16			129,15			22			212,98			28		
	PD22i	109,66			15			249,17			21			360,83			27		
	PGls	51,43			1525			168,71			5006			315,93			9372		

Tabuľka B.86: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 0.1% ($de = 0.001$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	0,16	0,11	0,19	577	389	692	1,10	0,77	1,35	4118	2778	4963	6,99	4,99	8,85	26185	17920	32412
APGls	0,30	0,21	0,37	635	445	793	2,12	1,50	2,64	4558	3192	5588	13,68	9,83	17,52	29376	20785	37367
DPG1	0,23	0,16	0,29	577	389	692	1,67	1,16	2,10	4128	2779	5016	10,65	7,40	13,36	26429	17764	32322
DPG1ls	0,17	0,13	0,20	349	265	399	1,15	0,90	1,34	2332	1819	2683	6,71	5,26	7,85	13678	10600	16194
DPG2	0,23	0,16	0,29	577	389	692	1,04	0,78	1,50	2566	1910	3844	2,37	1,47	7,75	5825	3688	18482
DPG2ls	0,17	0,14	0,21	350	265	399	0,70	0,54	0,98	1409	1097	1847	1,80	1,06	15,50	3629	2137	30656
DPG3	0,23	0,16	0,28	577	389	692	1,02	0,77	1,53	2531	1876	3844	2,63	1,51	9,02	6519	3852	21894
DPG3ls	0,17	0,14	0,24	351	265	449	0,71	0,53	0,94	1442	1091	1877	2,05	1,14	16,78	4172	2327	33386
PD11i	0,13	0,09	0,17	19	16	21	2,23	1,43	3,78	27	24	31	3,73	2,08	5,73	34	30	37
PD11ir	0,10	0,08	0,12	18	15	20	0,24	0,18	0,31	24	21	27	0,29	0,20	0,38	28	25	32
PD12i	0,31	0,22	0,40	20	17	22	4,38	2,72	8,69	31	27	35	7,35	4,15	11,61	37	33	42
PD13	3,79	2,65	5,47	17	16	18	5,24	3,61	7,28	24	23	25	8,04	5,97	10,81	30	29	31
PD21i	0,32	0,23	0,39	24	21	27	3,87	2,35	6,53	32	28	35	6,37	3,54	9,47	38	35	42
PD22i	1,06	0,74	1,51	22	19	26	7,47	4,63	12,98	33	28	39	11,42	6,45	17,21	40	35	45
PD23	3,25	2,21	4,27	15	14	15	4,65	3,32	6,47	21	21	22	7,56	5,74	9,64	28	27	28
PGls*	2,72	1,72	3,78	8262	5044	11393	24,51	12,54	41,27	74243	36780	119503	166,51	51,03	600,00	508K	156K	1,9M

Tabuľka B.88: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 1% ($de = 0.01$).

Presn. ϵ	Metóda	10^{-3}			10^{-6}			10^{-9}											
		čas		# iterácii	čas		# iterácii	čas		# iterácii									
		pr	min	max	pr	min	max	pr	min	max	pr	min	max						
	APG	0,10	0,09	0,12	269	225	310	0,73	0,60	0,86	1895	1553	2244	4,17	3,34	5,01	10669	8590	12966
	APGls	0,22	0,15	0,26	289	184	337	1,53	0,95	1,82	2050	1250	2453	8,77	4,84	10,63	11745	6520	14215
	DPG1	0,18	0,16	0,21	269	225	310	1,31	1,08	1,56	1895	1556	2244	7,53	5,94	9,08	10800	8572	13082
	DPG1ls	0,11	0,10	0,14	135	121	150	0,68	0,60	0,76	842	754	937	3,24	2,82	3,61	4059	3493	4562
	DPG2	0,19	0,16	0,22	269	225	310	1,19	0,92	1,48	1720	1342	2108	1,63	1,21	2,19	2339	1737	3174
	DPG2ls	0,11	0,10	0,13	135	121	150	0,48	0,29	0,72	590	325	869	0,66	0,45	0,91	807	520	1102
	DPG3	0,18	0,16	0,22	269	225	310	1,21	0,96	1,49	1744	1390	2108	1,84	1,37	2,34	2653	1977	3401
	DPG3ls	0,11	0,10	0,13	135	121	150	0,50	0,27	0,67	623	336	869	0,77	0,50	1,01	966	622	1239
	PD11i	0,17	0,13	0,22	15	14	17	1,24	1,05	1,47	21	21	24	1,86	1,49	2,25	28	27	30
	PD11ir	0,18	0,12	0,21	15	14	17	0,23	0,18	0,27	19	18	21	0,27	0,22	0,30	23	22	25
	PD12i	0,42	0,32	0,52	15	14	15	2,09	1,80	2,53	21	21	22	3,01	2,49	3,47	28	27	28
	PD13	5,19	5,10	5,49	15	15	16	7,42	7,14	7,64	22	21	22	9,58	9,24	9,97	28	27	29
	PD21i	0,46	0,37	0,56	16	15	17	2,80	2,33	3,37	22	21	23	4,13	3,37	4,97	28	27	30
	PD22i	1,24	1,02	1,54	14	14	15	4,52	3,99	5,20	21	20	22	6,13	5,06	7,15	27	27	28
	PD23	5,19	4,81	5,60	14	13	15	7,52	7,40	7,88	20	20	21	9,92	9,68	10,14	27	26	27
	PGLs	0,69	0,59	0,81	1452	1239	1722	3,03	2,36	4,11	6467	4996	8723	6,25	4,93	8,48	13352	10460	18178

Tabuľka B.89: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 5% ($de = 0.05$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																		
APG	0,45	0,36	0,50	318	253	348	3,23	2,55	3,58	2281	1809	2532	18,66	13,90	21,05	13214	9915	14831
APGls	0,99	0,54	1,23	352	192	432	7,10	3,64	8,90	2542	1312	3159	42,11	19,21	55,01	15117	6898	19558
DPG1	0,80	0,63	0,89	318	253	348	5,69	4,46	6,30	2279	1809	2515	33,27	24,44	37,59	13350	9864	15031
DPG1ls	0,32	0,26	0,35	105	86	115	1,69	1,38	1,90	566	462	629	6,83	5,32	7,51	2294	1790	2499
DPG2	0,80	0,63	0,89	318	253	348	5,14	2,89	6,32	2056	1165	2511	7,13	4,35	10,97	2850	1753	4403
DPG2ls	0,32	0,25	0,35	105	86	115	1,03	0,71	1,58	340	234	524	1,43	1,08	1,98	472	351	654
DPG3	0,80	0,63	0,87	318	253	348	5,20	3,04	6,29	2084	1215	2511	7,99	5,12	11,01	3203	2060	4435
DPG3ls	0,32	0,26	0,35	105	86	115	1,08	0,74	1,64	360	244	541	1,68	1,22	2,35	561	404	781
PD1li	0,62	0,52	0,78	14	14	15	3,02	2,37	3,33	21	20	21	4,52	3,59	5,10	27	26	28
PD1li ^r	0,53	0,43	0,62	14	13	15	0,84	0,68	0,96	18	18	19	0,96	0,77	1,09	22	22	23
PD12i	1,49	1,23	1,81	14	14	15	5,14	4,24	5,59	21	20	21	7,16	5,66	7,98	27	26	27
PD13	8,02	7,39	8,68	14	13	15	11,50	11,19	12,26	20	20	21	15,36	14,70	15,86	27	26	27
PD21i	1,58	1,31	2,04	14	13	15	7,36	5,68	8,08	21	20	21	10,77	8,50	12,11	27	26	28
PD22i	4,89	3,56	5,70	14	13	15	12,50	10,20	13,88	20	20	21	17,05	13,40	18,68	27	26	27
PD23	14,11	13,08	14,83	14	13	14	20,34	19,86	21,40	20	20	21	26,89	26,05	28,16	27	26	27
PGLs	1,31	1,02	1,51	705	546	811	4,85	3,43	5,66	2627	1861	3047	9,50	6,55	10,90	5146	3558	5896

Tabuľka B.90: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 10% ($de = 0.1$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	1,33	1,09	1,48	438	359	490	9,77	7,89	10,95	3240	2601	3635	59,27	47,17	66,35	19686	15642	22088
APGls	2,79			527			20,89			3965			129,47			24574		
DPG1	2,05			411			14,99			3017			92,11			18494		
DPG1ls	0,76	0,66	0,83	133	116	146	3,96	3,36	4,35	700	591	771	15,82	13,16	17,84	2804	2329	3162
DPG2	2,21	1,82	2,49	438	359	490	13,29	6,47	18,34	2645	1283	3635	17,85	10,36	25,17	3551	2058	4990
DPG2ls	0,76	0,67	0,83	133	116	146	2,26	1,47	3,98	397	254	704	3,10	2,31	4,74	543	400	837
DPG3	2,20	1,79	2,46	438	359	490	13,39	6,15	18,24	2668	1225	3635	20,67	11,15	26,57	4120	2212	5284
DPG3ls	0,76	0,66	0,84	133	116	146	2,37	1,66	4,06	419	291	718	3,78	2,79	5,32	669	491	941
PD11i	1,22	1,00	1,45	14	14	15	5,07	4,30	5,58	21	20	21	7,70	6,67	8,86	27	26	28
PD11ir	1,31	1,13	1,51	14	14	15	2,01	1,66	2,24	18	18	19	2,23	1,84	2,50	23	22	23
PD12i	3,28	2,75	3,76	14	13	15	9,33	8,41	10,43	20	20	21	13,11	11,78	14,90	27	26	27
PD13	8,19	7,63	8,71	14	13	14	12,15	11,46	12,64	20	19	21	16,12	15,50	17,32	26	26	28
PD21i	2,81	2,39	3,43	14	14	15	11,39	10,01	13,32	21	20	21	16,98	14,74	19,38	27	26	27
PD22i	8,77	7,60	10,51	13	13	14	21,50	18,52	23,95	20	19	20	28,87	25,44	33,20	26	26	27
PD23	14,78	13,91	15,73	14	13	14	21,81	20,62	22,38	20	19	20	28,68	27,66	31,09	26	26	28
PGls	3,53	2,76	4,04	968	759	1105	11,86	9,05	14,43	3261	2490	3960	22,74	17,76	27,43	6253	4880	7532

Tabuľka B.91: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 20% ($de = 0.2$).

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	3,86			652			29,30			4941			191,43			32241		
APGls	8,55			855			64,53			6456			449,41			45032		
DPG1	6,31			652			47,67			4940			317,30			32795		
DPG1ls	2,06	1,84	2,26	192	171	211	11,08	9,96	12,09	1039	932	1135	46,71	41,12	52,63	4385	3854	4943
DPG2	6,28	5,53	6,99	641	563	717	30,90	16,34	47,69	3161	1668	4887	46,72	27,22	64,34	4778	2783	6593
DPG2ls	2,06	1,85	2,26	192	171	211	5,98	3,72	8,27	557	344	772	8,80	5,75	11,07	818	531	1029
DPG3	6,27	5,51	6,99	641	563	717	27,76	16,92	47,56	2841	1721	4887	48,00	29,18	69,58	4914	2991	7133
DPG3ls	2,06	1,85	2,27	192	171	211	6,24	4,17	8,54	585	390	801	9,78	7,39	12,23	917	693	1148
PD11i	2,65	2,13	3,29	14	13	15	9,82	8,19	11,31	20	20	21	14,89	12,39	17,75	27	26	28
PD11ir	2,77	2,30	3,18	14	13	15	4,37	3,80	5,08	18	18	19	4,79	4,08	5,53	23	22	23
PD12i	7,17	6,07	7,92	14	13	14	17,60	15,52	19,20	20	20	21	24,60	20,90	27,34	26	26	27
PD13	8,28	7,64	8,76	14	13	14	12,09	11,79	12,77	20	20	21	16,89	15,45	23,60	28	26	37
PD21i	5,28	4,21	6,07	14	13	15	18,64	16,00	20,47	21	20	22	28,12	23,75	32,25	27	26	28
PD22i	17,96	14,13	20,92	14	13	15	38,86	34,04	43,08	20	20	21	53,24	44,72	59,29	27	26	27
PD23	15,08	14,26	15,94	14	13	15	21,67	21,13	22,72	20	20	21	30,51	27,72	55,62	28	26	46
PGls	12,79			1885			45,17			6663			120,47			17773		

Tabuľka B.92: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 40% ($de = 0.4$).

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	14,78			1082			116,61			8568			774,30			56932								
DPG1ls	7,76			329			50,80			2160			227,01			9657								
DPG2	23,89			1082			116,78			5320			201,78			9182								
DPG2ls	7,49	6,29	8,81	323	275	382	23,94	15,08	31,11	1034	643	1333	37,99	29,73	48,75	1640	1285	2144						
DPG3	23,76			1082			126,19			5738			172,14			7832								
DPG3ls	7,50	6,33	8,88	323	275	382	25,25	18,16	33,53	1094	779	1438	41,70	32,58	49,69	1808	1395	2184						
PD1i	7,85	4,88	10,35	14	13	15	23,72	19,02	29,53	21	20	21	35,35	29,97	42,36	27	26	28						
PD1ir	7,82	5,73	10,32	14	13	15	12,60	8,97	16,09	19	18	20	13,73	9,96	17,55	23	22	24						
PD12i	21,53	16,55	26,40	14	13	14	49,18	42,65	56,64	20	19	20	66,58	57,76	79,43	26	26	27						
PD13*	8,05	7,58	8,57	14	13	14	11,88	11,42	12,41	20	19	20	42,60	15,27	60,83	58	26	79						
PD21i	10,91	8,78	13,11	14	13	15	36,04	30,55	44,42	21	20	22	55,39	46,33	66,03	27	26	29						
PD22i	56,47			14			109,13			20			144,77			26								
PD23*	14,69	14,22	15,70	14	14	15	21,45	20,35	22,46	21	20	21	52,46	28,20	61,60	42	27	49						

Tabuľka B.93: Vplyv riedkosti matice dát: Porovnanie metód riešenia na 50-tich úlohách (1.7) s hustotou zaplnenia matice dát 100% ($de = 1$).

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
DPG1ls	18,95			917			118,04			5729			936,79			45499		
DPG2	56,82			2890			586,74			29854			684,11			34806		
DPG2ls	20,28	17,53	26,52	974	840	1275	99,85	37,41	130,05	4822	1795	6301	131,25	105,19	165,42	6338	5060	8003
DPG3	58,56			2978			681,26			34679			765,15			38948		
DPG3ls	21,67	18,74	26,58	1046	902	1282	113,11	49,29	152,58	5471	2376	7398	160,49	132,40	216,40	7763	6413	10461
PD1i	8,07	4,15	11,70	14	14	15	27,46	19,70	31,95	21	20	22	41,18	33,06	50,00	27	27	28
PD1ir	8,95	5,08	12,82	14	14	15	19,65	13,66	24,27	20	19	21	21,05	14,58	25,65	24	23	25
PD12i	26,89	20,96	33,68	14	13	14	61,86	50,65	74,16	20	19	21	83,14	70,49	98,03	26	26	27
PD13	8,05	7,23	8,69	14	13	14	11,89	11,46	12,42	20	19	21						
PD21i	14,15	9,42	18,95	14	14	15	49,87	39,45	60,55	21	20	22	75,61	63,63	89,57	27	27	28
PD22i	68,49			14			131,45			20			177,61			27		
PD23	14,52	13,92	15,69	14	14	15	22,03	20,29	29,07	21	20	26						

Tabuľka B.94: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 100$ a $n = 1000$.

Metóda	Presn. ϵ						10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	7,8E-4	4,6E-4	3,5E-3	62	38	93	3,4E-3	2,0E-3	7,6E-3	316	190	477	1,3E-2	7,6E-3	2,3E-2	1339	721	2093						
APGls	1,1E-3	5,5E-4	5,7E-3	42	26	64	4,4E-3	2,4E-3	1,0E-2	204	115	330	1,7E-2	9,8E-3	2,8E-2	830	460	1445						
DPG1	1,2E-3	6,4E-4	5,0E-3	62	38	93	5,4E-3	3,1E-3	1,2E-2	320	191	494	2,3E-2	1,5E-2	3,8E-2	1456	802	2422						
DPG1ls	8,0E-4	4,5E-4	6,0E-3	28	20	37	2,9E-3	1,8E-3	9,6E-3	120	83	169	9,6E-3	6,3E-3	1,9E-2	431	276	664						
DPG2	1,3E-3	6,4E-4	5,1E-3	62	38	93	4,0E-3	2,1E-3	9,0E-3	216	119	385	6,6E-3	3,8E-3	1,4E-2	372	217	674						
DPG2ls	8,2E-4	4,5E-4	5,9E-3	28	20	37	2,3E-3	1,4E-3	9,5E-3	89	61	148	3,7E-3	2,3E-3	1,4E-2	148	101	246						
DPG3	1,2E-3	6,3E-4	4,9E-3	62	38	93	3,9E-3	2,3E-3	7,9E-3	223	128	364	6,9E-3	3,8E-3	1,3E-2	403	222	688						
DPG3ls	8,0E-4	4,5E-4	5,7E-3	28	20	37	2,3E-3	1,4E-3	7,4E-3	92	63	159	3,8E-3	2,5E-3	9,2E-3	159	104	249						
PD11i	5,4E-3	3,6E-3	2,5E-2	12	11	13	1,1E-2	7,4E-3	3,2E-2	18	18	19	1,5E-2	1,1E-2	3,6E-2	25	24	26						
PD11ir	5,9E-3	3,9E-3	2,8E-2	12	11	13	7,3E-3	4,8E-3	3,2E-2	14	13	15	7,8E-3	5,3E-3	3,5E-2	18	16	20						
PD12i	3,2E-2	2,4E-2	4,8E-2	12	11	13	6,7E-2	4,9E-2	9,3E-2	18	18	19	9,2E-2	7,0E-2	1,3E-1	25	24	26						
PD13	8,0E-3	6,4E-3	2,8E-2	12	11	12	1,2E-2	1,0E-2	3,2E-2	18	18	19	1,6E-2	1,4E-2	3,5E-2	25	24	25						
PD21i	1,3E-2	9,1E-3	3,0E-2	11	11	12	2,7E-2	2,0E-2	4,4E-2	18	17	19	3,6E-2	2,7E-2	5,4E-2	24	24	25						
PD22i	5,0E-2	3,4E-2	6,7E-2	11	11	12	1,0E-1	7,3E-2	1,4E-1	18	17	19	1,3E-1	9,8E-2	2,0E-1	24	24	25						
PD23	9,0E-3	7,9E-3	2,0E-2	11	11	12	1,4E-2	1,2E-2	2,5E-2	18	17	19	1,9E-2	1,7E-2	3,0E-2	24	24	25						
PGls	1,3E-3	6,8E-4	5,1E-3	71	42	107	6,3E-3	2,5E-3	1,8E-2	422	168	1287	1,5E-2	5,6E-3	4,1E-2	1063	390	3004						

Tabuľka B.95: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 500$ a $n = 5000$.

Metóda	10 ⁻³						10 ⁻⁶						10 ⁻⁹					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	1,1E-2	9,1E-3	1,4E-2	106	85	132	6,4E-2	5,0E-2	7,8E-2	613	478	740	3,2E-1	2,2E-1	4,0E-1	3084	2100	3846
APGls	1,4E-2	1,1E-2	1,8E-2	79	62	103	7,7E-2	5,4E-2	1,0E-1	437	311	574	3,7E-1	2,6E-1	5,2E-1	2138	1451	2979
DPG1	1,6E-2	1,3E-2	2,0E-2	106	85	132	9,4E-2	7,4E-2	1,1E-1	614	479	743	4,8E-1	3,3E-1	6,1E-1	3153	2155	4002
DPG1ls	1,0E-2	9,1E-3	1,3E-2	54	47	63	5,2E-2	4,2E-2	6,0E-2	279	220	328	2,4E-1	1,7E-1	2,8E-1	1280	935	1507
DPG2	1,6E-2	1,3E-2	2,0E-2	106	85	132	7,4E-2	4,3E-2	9,6E-2	479	280	621	1,2E-1	7,4E-2	1,7E-1	798	480	1089
DPG2ls	1,0E-2	8,5E-3	1,3E-2	54	47	63	3,6E-2	2,9E-2	4,9E-2	195	159	272	6,8E-2	4,7E-2	1,2E-1	360	261	683
DPG3	1,6E-2	1,3E-2	2,1E-2	106	85	132	7,4E-2	4,4E-2	9,6E-2	482	291	636	1,4E-1	9,3E-2	1,8E-1	891	613	1161
DPG3ls	1,1E-2	8,4E-3	1,4E-2	54	47	63	3,8E-2	2,7E-2	5,1E-2	201	152	277	7,4E-2	4,9E-2	1,1E-1	397	275	610
PD11i	8,3E-2	6,1E-2	1,0E-1	13	12	13	2,8E-1	1,9E-1	3,7E-1	19	19	20	4,0E-1	2,7E-1	5,3E-1	26	25	26
PD11ir	8,2E-2	6,2E-2	1,0E-1	13	12	13	1,0E-1	7,6E-2	1,4E-1	16	15	16	1,0E-1	7,7E-2	1,4E-1	20	19	21
PD12i	1,9E-1	1,6E-1	2,4E-1	13	13	14	6,1E-1	4,4E-1	8,0E-1	19	19	20	8,4E-1	6,0E-1	1,1E+0	26	26	27
PD13	2,2E-1	1,9E-1	2,8E-1	13	12	14	3,3E-1	2,9E-1	3,9E-1	19	19	20	4,9E-1	4,3E-1	5,7E-1	26	25	26
PD21i	2,3E-1	1,6E-1	2,9E-1	13	12	13	7,4E-1	5,1E-1	1,0E+0	19	18	19	1,0E+0	7,1E-1	1,4E+0	25	25	26
PD22i	4,9E-1	4,0E-1	6,1E-1	13	12	13	1,3E+0	9,2E-1	1,6E+0	19	19	20	1,7E+0	1,3E+0	2,2E+0	26	25	26
PD23	2,4E-1	2,0E-1	2,8E-1	13	12	13	3,5E-1	3,1E-1	4,1E-1	19	19	20	5,1E-1	4,4E-1	6,0E-1	26	25	26
PGLs	3,3E-2	2,3E-2	4,4E-2	250	182	317	2,4E-1	1,5E-1	4,0E-1	1860	1139	3038	7,1E-1	3,1E-1	3,3E+0	5470	2464	25571

Tabuľka B.96: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmerni $m = 1000$ a $n = 10000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
Metóda	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	3,1E-2	2,5E-2	3,6E-2	142	113	164	1,9E-1	1,6E-1	2,2E-1	858	709	987	1,0E+0	7,3E-1	1,2E+0	4541	3284	5564
APGls	4,2E-2	3,4E-2	5,2E-2	110	90	133	2,4E-1	2,0E-1	2,9E-1	653	549	794	1,2E+0	9,8E-1	1,6E+0	3332	2641	4348
DPG1	4,7E-2	3,8E-2	5,4E-2	142	113	164	2,8E-1	2,4E-1	3,3E-1	857	709	990	1,5E+0	1,1E+0	1,8E+0	4582	3333	5397
DPG1ls	3,1E-2	2,7E-2	3,7E-2	75	64	84	1,6E-1	1,4E-1	1,8E-1	404	359	456	7,7E-1	6,3E-1	9,4E-1	1964	1584	2343
DPG2	4,7E-2	3,7E-2	5,4E-2	142	113	164	2,6E-1	1,9E-1	3,1E-1	780	579	956	4,1E-1	3,2E-1	5,5E-1	1225	960	1645
DPG2ls	3,1E-2	2,6E-2	3,7E-2	75	64	84	1,2E-1	8,4E-2	1,6E-1	289	205	404	2,0E-1	1,5E-1	2,8E-1	510	375	702
DPG3	4,7E-2	3,8E-2	5,4E-2	142	113	164	2,6E-1	2,0E-1	3,1E-1	791	587	956	4,5E-1	3,6E-1	5,6E-1	1351	1072	1683
DPG3ls	3,1E-2	2,7E-2	3,6E-2	75	64	84	1,2E-1	9,0E-2	1,6E-1	300	223	404	2,4E-1	1,8E-1	3,4E-1	597	461	864
PD11i	1,7E-1	1,4E-1	2,8E-1	13	13	14	8,8E-1	6,4E-1	1,2E+0	20	19	20	1,2E+0	9,6E-1	1,7E+0	26	26	27
PD11ir	1,8E-1	1,4E-1	2,8E-1	13	13	14	2,4E-1	1,7E-1	3,2E-1	16	16	17	2,5E-1	1,7E-1	3,3E-1	20	20	21
PD12i	3,8E-1	2,9E-1	4,8E-1	13	13	14	1,5E+0	1,2E+0	2,0E+0	20	19	20	2,1E+0	1,7E+0	2,8E+0	26	26	27
PD13	8,3E-1	6,4E-1	1,1E+0	13	13	14	1,2E+0	9,9E-1	1,6E+0	20	19	21	1,8E+0	1,4E+0	2,2E+0	26	26	27
PD21i	3,5E-1	2,9E-1	4,5E-1	13	13	14	1,5E+0	1,2E+0	2,1E+0	20	19	20	2,1E+0	1,6E+0	2,9E+0	26	25	26
PD22i	7,8E-1	6,2E-1	1,0E+0	13	13	14	2,4E+0	2,0E+0	3,1E+0	20	19	20	3,2E+0	2,6E+0	4,2E+0	26	25	27
PD23	8,4E-1	6,7E-1	1,1E+0	13	13	14	1,3E+0	9,8E-1	1,6E+0	20	19	20	1,8E+0	1,5E+0	2,3E+0	26	26	26
PGLs	1,3E-1	1,0E-1	1,6E-1	463	367	585	8,9E-1	5,8E-1	1,3E+0	3300	2111	5031	2,7E+0	1,4E+0	7,2E+0	10099	5161	26938

Tabuľka B.97: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 5000$ a $n = 50000$.

Metóda	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
APG	1,9E-1	1,6E-1	2,2E-1	278	232	324	1,3E+0	1,1E+0	1,6E+0	1865	1557	2174	8,0E+0	6,4E+0	9,4E+0	10915	8685	12785
APGls	3,8E-1	2,7E-1	4,9E-1	266	190	343	2,5E+0	1,7E+0	3,4E+0	1787	1237	2403	1,5E+1	9,6E+0	2,0E+1	10482	6838	14245
DPG1	3,8E-1	3,2E-1	4,4E-1	278	232	324	2,5E+0	2,1E+0	3,0E+0	1865	1557	2174	1,5E+1	1,2E+1	1,7E+1	10934	8704	12856
DPG1ls	2,4E-1	2,1E-1	2,6E-1	156	137	172	1,4E+0	1,3E+0	1,6E+0	963	862	1042	7,8E+0	6,7E+0	8,6E+0	5221	4493	5748
DPG2	3,8E-1	3,1E-1	4,4E-1	278	232	324	2,5E+0	2,0E+0	3,0E+0	1854	1499	2174	4,8E+0	3,2E+0	6,3E+0	3498	2330	4654
DPG2ls	2,4E-1	2,1E-1	2,6E-1	156	137	172	1,1E+0	7,9E-1	1,5E+0	706	523	990	1,8E+0	1,4E+0	2,3E+0	1187	955	1563
DPG3	3,8E-1	3,1E-1	4,4E-1	278	232	324	2,5E+0	2,1E+0	3,0E+0	1856	1548	2174	4,9E+0	3,2E+0	6,4E+0	3575	2393	4724
DPG3ls	2,4E-1	2,1E-1	2,6E-1	156	137	172	1,1E+0	8,0E-1	1,5E+0	739	537	990	2,2E+0	1,5E+0	2,9E+0	1453	1002	1942
PD11i	6,0E-1	4,5E-1	8,8E-1	14	14	15	5,7E+0	4,3E+0	9,7E+0	20	20	21	9,2E+0	7,1E+0	1,3E+1	27	26	28
PD11ir	6,2E-1	4,7E-1	9,0E-1	14	14	15	7,7E-1	6,4E-1	1,1E+0	17	17	18	9,0E-1	7,4E-1	1,2E+0	22	21	23
PD12i	1,3E+0	9,2E-1	1,6E+0	15	14	16	8,3E+0	6,7E+0	9,7E+0	21	21	22	1,3E+1	1,0E+1	1,7E+1	28	27	29
PD13	3,1E+1	2,4E+1	3,6E+1	15	14	15	4,4E+1	3,3E+1	5,1E+1	21	21	22	6,3E+1	5,0E+1	7,3E+1	28	27	28
PD21i	1,8E+0	1,4E+0	3,6E+0	14	14	15	1,5E+1	1,1E+1	2,5E+1	21	20	22	2,3E+1	1,8E+1	3,4E+1	27	27	28
PD22i	4,6E+0	3,6E+0	5,9E+0	15	14	16	2,0E+1	1,6E+1	2,3E+1	21	20	22	3,0E+1	2,3E+1	3,7E+1	27	27	29
PD23	3,0E+1	2,2E+1	3,6E+1	14	14	15	4,3E+1	3,3E+1	5,1E+1	21	20	21	6,2E+1	4,8E+1	7,1E+1	27	26	28
PGLs	1,6E+0	1,3E+0	1,9E+0	1854	1490	2203	1,2E+1	8,3E+0	1,8E+1	13778	9794	21501	3,9E+1	2,5E+1	7,3E+1	46518	29336	86674

Tabuľka B.98: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 10000$ a $n = 100000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácií			čas			# iterácií			čas			# iterácií		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																		
APG	5,1E-1	4,1E-1	5,8E-1	392	314	440	3,5E+0	2,8E+0	4,0E+0	2727	2183	3085	2,2E+1	1,7E+1	2,6E+1	16735	12851	19492
APGls	1,1E+0	7,1E-1	1,3E+0	391	266	473	7,3E+0	4,7E+0	8,7E+0	2729	1791	3322	4,5E+1	2,8E+1	5,5E+1	16906	10469	21100
DPG1	1,0E+0	8,1E-1	1,1E+0	392	314	440	7,0E+0	5,6E+0	7,9E+0	2727	2183	3085	4,3E+1	3,3E+1	5,0E+1	16744	12863	19559
DPG1ls	6,4E-1	5,5E-1	6,9E-1	222	188	237	4,1E+0	3,5E+0	4,5E+0	1437	1224	1551	2,3E+1	2,0E+1	2,6E+1	8150	6871	8920
DPG2	1,0E+0	8,1E-1	1,1E+0	392	314	440	7,0E+0	5,6E+0	8,0E+0	2727	2183	3085	1,5E+1	8,9E+0	2,3E+1	5996	3450	8868
DPG2ls	6,5E-1	5,5E-1	6,9E-1	222	188	237	3,5E+0	2,0E+0	4,4E+0	1225	686	1531	5,4E+0	4,1E+0	7,3E+0	1886	1442	2569
DPG3	1,0E+0	8,0E-1	1,1E+0	392	314	440	7,0E+0	5,6E+0	7,9E+0	2727	2183	3085	1,5E+1	8,9E+0	2,3E+1	6054	3495	8934
DPG3ls	6,4E-1	5,5E-1	6,9E-1	222	188	237	3,6E+0	2,3E+0	4,3E+0	1257	790	1531	6,1E+0	4,3E+0	7,3E+0	2147	1511	2571
PD11i	1,3E+0	8,9E-1	1,7E+0	15	14	17	1,6E+1	1,2E+1	2,6E+1	21	21	23	2,9E+1	2,2E+1	4,3E+1	28	27	30
PD11ir	1,1E+0	8,2E-1	1,4E+0	15	14	17	1,6E+0	1,3E+0	1,9E+0	19	18	21	2,2E+0	1,8E+0	2,7E+0	23	22	25
PD12i	3,1E+0	2,3E+0	4,1E+0	16	15	17	2,5E+1	2,0E+1	3,2E+1	23	21	24	4,5E+1	3,5E+1	6,1E+1	29	28	30
PD13	1,7E+2			15			2,4E+2			21			3,5E+2			28		
PD21i	3,9E+0			15			4,2E+1			21			8,3E+1			28		
PD22i	1,7E+1			17			7,0E+1			23			1,2E+2			29		
PD23	1,7E+2			15			2,4E+2			21			3,4E+2			27		
PGls	5,9E+0			3610			4,5E+1			27748			1,4E+2			86739		

Tabuľka B.99: Škálovateľnosť. Vplyv rozmernosti úloh: Porovnanie metód riešenia na 50-tich úlohách (1.7) s rozmermi $m = 50000$ a $n = 500000$.

Presn. ϵ	10^{-3}						10^{-6}						10^{-9}					
	čas			# iterácii			čas			# iterácii			čas			# iterácii		
	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max	pr	min	max
Metóda																		
APG	7,1E+0		867				5,2E+1		6314	3,5E+2		43018						
APGls	1,4E+1		959				1,0E+2		6992	7,2E+2		48874						
DPG1	1,2E+1		867				8,8E+1		6314	6,0E+2		43025						
DPG1ls	8,0E+0		501				5,5E+1		3516	3,5E+2		22008						
DPG2	1,2E+1		867				8,8E+1		6314	3,1E+2		22029						
DPG2ls	7,6E+0	6,7E+0	8,3E+0	476	415	515	5,2E+1	3,3E+1	5,7E+1	8,4E+1	5,7E+1	1,3E+2	5263	3545	7896			
DPG3	1,2E+1		867				8,8E+1		6314	3,1E+2		22111						
DPG3ls	7,6E+0	6,7E+0	8,2E+0	476	415	515	5,2E+1	3,5E+1	5,7E+1	8,9E+1	6,5E+1	1,3E+2	5576	4099	7925			
PD1li	1,1E+1		25				2,0E+2		32	4,5E+2		38						
PD1li ^r	6,5E+0	5,2E+0	8,5E+0	24	20	29	1,3E+1	1,0E+1	1,5E+1	1,8E+1	1,3E+1	2,4E+1	32	28	38			
PD12i	2,8E+1		23				2,9E+2		32	7,6E+2		39						

