Linköping Studies in Science and Technology. Dissertations.
No. 1561

Large-Scale Optimization Methods with Application to Design of Filter Networks

Spartak Zikrin

Linköping University

Department of Mathematics, Division of Optimization
Linköping University, SE-581 83 Linköping, Sweden

Linköping 2014

Cover picture: “Band-pass filter of order 2”.

Linköping Studies in Science and Technology. Dissertations. No. 1561

**Large-Scale Optimization Methods with
Application to Design of Filter Networks**

Copyright © 2014 Spartak Zikrin

*Division of Optimization
Department of Mathematics
Linköping university
SE-581 83, Linköping, Sweden
spartak.zikin@liu.se
www.mai.liu.se/opt*

Typeset by the author in L^AT_EX2e documentation system.

ISSN 0345-7524

ISBN 978-91-7519-456-1

Printed by LiU-Tryck, Linköping, Sweden 2014

Abstract

Nowadays, large-scale optimization problems are among those most challenging. Any progress in developing methods for large-scale optimization results in solving important applied problems more effectively. Limited memory methods and trust-region methods represent two efficient approaches used for solving unconstrained optimization problems. A straightforward combination of them deteriorates the efficiency of the former approach, especially in the case of large-scale problems. For this reason, the limited memory methods are usually combined with a line search. We develop new limited memory trust-region algorithms for large-scale unconstrained optimization. They are competitive with the traditional limited memory line-search algorithms.

In this thesis, we consider applied optimization problems originating from the design of filter networks. Filter networks represent an efficient tool in medical image processing. It is based on replacing a set of dense multidimensional filters by a network of smaller sparse filters called sub-filters. This allows for improving image processing time, while maintaining image quality and the robustness of image processing.

Design of filter networks is a nontrivial procedure that involves three steps: 1) choosing the network structure, 2) choosing the sparsity pattern of each sub-filter and 3) optimizing the nonzero coefficient values. So far, steps 1 and 2 were mainly based on the individual expertise of network designers and their intuition. Given a sparsity pattern, the choice of the coefficients at stage 3 is related to solving a weighted nonlinear least-squares problem. Even in the case of sequentially connected filters, the resulting problem is of a multilinear least-squares (MLLS) type, which is a non-convex large-scale optimization problem. This is a very difficult global optimization problem that may have a large number of local minima, and each of them is singular and non-isolated. It is characterized by a large number of decision variables, especially for 3D and 4D filters.

We develop an effective global optimization approach to solving the MLLS problem that reduces significantly the computational time. Furthermore, we develop efficient methods for optimizing sparsity of individual sub-filters in filter networks of a more general structure. This approach offers practitioners a means of finding a proper trade-off between the image processing quality and time. It allows also for improving the network structure, which makes automated some stages of designing filter networks.

Populärvetenskaplig sammanfattning

Inom modern medicin används avancerade tekniker, som datortomografi (CT) och magnetresonanstomografi (MRI), för att skapa tre-dimensionella bilder av delar av människokroppen. För att underlätta analysen av bilderna är det ofta önskvärt att förbättra deras kvalitet, genom att ta bort störningar i bilden eller förbättra dess kontrast, eller genom att förändra bilden för att underlätta för det mänskliga ögat att urskilja intressanta avvikelser i bilden.

Förbättring av bilders kvalitet görs genom att de på olika sätt filtreras, vilket innebär att de systematiskt modifieras med hjälp av matematiska algoritmer. Dessa algoritmers utformning beror på bildernas ursprung och på vilket sätt bilden ska förbättras.

Ett sätt att effektivisera filtreringen är att göra den med ett filternät, vilket innebär att filtreringen delas upp i flera och enklare steg. För att filternätet ska uppfylla sitt syfte så väl som möjligt måste dess konstruktion optimeras, vilket utmynnar i matematiska optimeringsproblem som har en speciell struktur och som är mycket svårlösta.

Den här avhandlingen presenterar effektiva metoder för optimeringsproblem av denna typ. Detta angreppssätt ger tillämpare ett sätt att hitta en lämplig avvägning mellan bilders kvalitet och beräkningstid.

Vi har också utvecklat nya optimeringsmetoder för att minimera en generell funktion med många variabler. De är kombinationer av *limited memory*-tekniker, som har låg kostnad för beräkningstid och minneslagring, och *trust-region*-tekniker, vilka innebär att man kan använda en enkel modell istället för funktionen i ett begränsad område där man litar på sin modell.

Acknowledgements

First of all, I would like to thank my supervisor Oleg Burdakov, whose help was invaluable in many ways. It was a great privilege to work with such a prominent mathematician, who not only introduced me to the area of nonlinear optimization and guided in writing this thesis but provided any support in everyday life. Special greetings are to his wonderful family.

I also thank my co-supervisors Hans Knutsson and Mats Andersson for introducing me into the area of medical image analysis. Having only a mathematical background, this was a completely new subject for me. Our numerous discussions about filter networks resulted in two joint papers.

One more paper was written in collaboration with yet another highly distinguished mathematician Ya-xiang Yuan. I thank him and members of his research group who provided excellent hospitality during my visit to the Chinese Academy of Sciences.

I extend further my gratitude to all members of the optimization group. All of you contributed in various ways to this thesis, although we often had different subjects of research.

Furthermore, I want to thank all other colleagues, former and present, at the Department of Mathematics, for a friendly working atmosphere. With some of you, including Aysajan, Henry, Jolanta, Anna, Sonia, Alexandra, I share great memories out of work.

I acknowledge contribution of all my former mentors in mathematics, including Bagdat Tynykenova and Sergey Gashkov.

I send greeting to all corners of the world to international friends I met in Linköping and to all members of ISA. Extra greetings are to the Russian-speaking friends and my countrymen from Kazakhstan I met in Scandinavia. You made my days brighter during the PhD studies.

Finally, I owe a lot of thanks to my caring mother Olga, who always supported my decisions, morally and until some years ago financially, which was definitely not an easy task. My father stays with me in memory, the gens I inherited from him brought me to mathematics.

Linköping, January 21, 2014

Spartak Zikrin

List of Papers

The following papers are appended and will be referred to by their roman numerals.

- I.** O. Burdakov, L. Gong, Y. Yuan, S. Zikrin, On Efficiently Combining Limited Memory and Trust-Region Techniques, submitted to *SIAM Journal on Optimization*.
Available as: *Technical Report LiTH-MAT-R-2013/13-SE*, Department of Mathematics, Division of Optimization, Linköping University, 2013.
- II.** M. Andersson, O. Burdakov, H. Knutsson, S. Zikrin, Global Search Strategies for Solving Multilinear Least-squares Problems, *Sultan Qaboos University Journal for Science* 17, pp. 12-21, 2012.
Available as: *Technical Report LiTH-MAI-R-2011/17-SE*, Department of Mathematics, Division of Optimization, Linköping University, 2011.
- III.** M. Andersson, O. Burdakov, H. Knutsson, S. Zikrin, Sparsity Optimization in Design of Multidimensional Filter Networks, submitted to *Journal on Optimization and Engineering*.
Available as: *Technical Report LiTH-MAI-R-2013/16-SE*, Department of Mathematics, Division of Optimization, Linköping University, 2013.

Acronyms and Abbreviations

ALS	Alternating least-squares method
BFGS	Broyden-Fletcher-Goldfarb-Shanno quasi-Newton update
CG	Conjugate gradients
FIR	Finite impulse response
L-BFGS	Limited memory BFGS update
MLLS	Multilinear least-squares problem
OMP	Orthogonal matching pursuit
OLS	Orthogonal least-squares pursuit
SR1	Symmetric-rank-one quasi-Newton update
SVD	Singular value decomposition

Contents

Abstract	i
Sammanfattning	iii
Acknowledgements	v
List of Papers	vii
Acronyms and Abbreviations	ix
1 Introduction	1
1 Outline	1
2 Contributions	1
2.1 Papers	1
2.2 Contributions of co-authors	2
2.3 Conference presentations	3
2 Unconstrained Optimization	5
1 Introduction	5
1.1 Optimality conditions	5
1.2 Line-search methods	7
1.3 Trust-region methods	10
2 Conjugate gradient methods	14
2.1 Linear conjugate gradient methods	14
2.2 Nonlinear conjugate gradient methods	16

**Large-Scale Optimization Methods with
Application to Design of Filter Networks**

3	Quasi-Newton methods	18
4	Large-scale optimization methods	20
4.1	Line-search inexact Newton methods	21
4.2	Trust-region inexact Newton methods	21
4.3	Limited memory quasi-Newton updates	22
5	Nonlinear least-squares problems	25
6	Cardinality-constrained linear least-squares problem	26
6.1	Greedy pursuit methods	27
6.2	Convex relaxation methods	30
7	Multiobjective optimization	32
3	Optimal Design of Filter Networks	33
1	Design of FIR filters	33
2	Filter networks	37
3	Multilinear least-squares problem	38
3.1	Global search approach	40
3.2	Continuation approach	42
4	Summary of Papers	45
	Paper I - On Efficiently Combining Limited Memory and Trust-Region Techniques	53

Appended Papers

1	Introduction	56
2	Solving the trust-region subproblem	58
2.1	Spectrum of the quasi-Newton matrix	58
2.2	Trust-region subproblem in the Euclidean norm	59
2.3	Eigenvalue-based decomposition of the model function	61
2.4	Trust-region subproblem in new norms	61
2.5	Algorithm	64

CONTENTS

3	Convergence Analysis	64
4	Implementation details for L-BFGS	67
4.1	Uniform presentation of eigenvalue-based solutions	67
4.2	Model function evaluation	68
4.3	Updating Hessian approximation	69
4.4	Numerical stability	69
4.5	Computational complexity	70
4.6	Computing the quasi-Newton step	70
4.7	Improved versions of existing limited memory trust-region approaches	71
5	Numerical tests	74
6	Conclusions	81
 Paper II - Global Search Strategies for Solving Multilinear Least-squares Problems		85
1	Introduction	87
2	Problem reformulation	88
3	Theoretical background for global search	91
4	Global search algorithm	94
5	Numerical experiments	95
6	Conclusions	96
 Paper III - Sparsity Optimization in Design of Multidimensional Filter Networks		99
1	Introduction	101
2	Design of sequentially connected sub-filters	103
2.1	Cardinality-constrained MLLS	104
2.2	Bi-criteria optimization problem	107
3	Design of layered filter networks	109
4	Numerical Experiments	111
5	Conclusions	116

Introduction

For many real world applications, their success is related to design of efficient methods for solving large-scale optimization problems. Such methods typically require to be modest in computational time and memory storage. In this thesis, we consider large-scale unconstrained optimization problems and develop limited memory trust-region methods. We also consider applied optimization problems originating from the design of filter networks. Filter networks is a powerful tool in medical image analysis. It allows for reducing the image processing time, while maintaining sufficiently high image quality.

1 Outline

The material in this thesis is organized as follows.

In Chapter 2, we present a brief introduction into the area of unconstrained optimization, with special emphasis on methods for solving large-scale problems. This chapter provides background for non-specialists in optimization for reading Papers I–III.

In Chapter 3, we provide background material related to design of filter networks for reading Papers II and III . This material is written for specialists in optimization who are not familiar with main concepts in filter design. Also, this chapter contains some of the obtained results which have not been published.

In Chapter 4, we formulate main conclusions of this thesis and discuss future work.

2 Contributions

2.1 Papers

This thesis contains three papers, of which one have been published in a journal and two have been submitted and available as technical reports. Here follows

a short summary of these papers. The first paper concerns the area of large-scale unconstrained optimization problems, and the last two papers concern developing efficient optimization methods for design of filter networks.

Paper I. On Efficiently Combining Limited Memory and Trust-Region Techniques

Develops combinations of limited memory and trust-region techniques for solving large-scale optimization problems and studies their theoretical properties. The numerical experiments demonstrate their efficiency.

Paper II. Global Search Strategies for Solving Multilinear Least-squares Problems

Develops an effective global optimization approach to solving the multilinear least-squares problem. We have proved the correctness of this approach theoretically and demonstrated its efficiency on test problems originating from design of filter networks.

Paper III. Sparse Optimization of Filter Networks

Develops computationally efficient methods for optimizing sparsity of individual sub-filters in filter networks. Our approach offers practitioners a means of finding a proper trade-off between the image processing quality and time. It allows also for improving the network structure, which makes automated some stages of designing filter networks.

2.2 Contributions of co-authors

All papers are co-authored with Oleg Burdakov, whose role was to supervise the work and to some degree suggest directions of research. In Paper I, Ya-xiang Yuan contributed with establishing the convergence of our algorithms, and Lujin Gong contributed with developing a preliminary version of the MATLAB implementation of the proposed algorithms. Mats Andersson and Hans Knutsson supervised the application part of the thesis work. As co-authors of Papers II and III, they formulated problems related to the design of filter networks.

I wrote a draft of Theorem 1 and its proof in Paper II. I also wrote drafts for Paper I and Paper III. The co-authors helped me to improve these drafts and write final versions. I implemented the most of our algorithms and performed all numerical experiments.

2.3 Conference presentations

Parts of this thesis have been presented by the author at the following international conferences.

1. **The 3rd International Conference on Optimization Methods and Software**, Chania, Greece, May 13-17, 2012. Presented parts of Paper III.
2. **The 21st International Symposium on Mathematical Programming**, Berlin, Germany, August 19-24, 2012. Presented parts of Paper III.
3. **Conference on Inverse Problems and Applications**, Linköping, Sweden, April 2-6, 2013. Presented parts of Paper III.
4. **The 4th International Conference on Continuous Optimization** (organized by *the Mathematical Optimization Society*), Lisbon, Caparica, Portugal, July 27 - August 1, 2013. Presented parts of Paper I.

Oleg Burdakov also presented parts of our papers at the following international conferences.

1. **The 3rd Nordic Optimization Symposium**, Stockholm, Sweden, March 13-14, 2009. Presented parts of Paper II.
2. An invited talk at **International Conference on Engineering and Computational Mathematics**, Hong Kong, May 27-29, 2009. Presented parts of Paper II.
3. An invited plenary talk at **The 2nd International Conference on Optimization and Numerical Analysis**, Muscat, Oman, January 3-6, 2011. Presented parts of Paper II.
4. **The 2nd World Congress on Global Optimization in Engineering and Science**, Crete, Greece, May 13-17, 2011. Presented parts of Paper II and Paper III.
5. **International Conference on Numerical Computations**, Falerna, Italy, June 17-23, 2013. Presented parts of Paper I.
6. **The 7th Moscow International Conference on Operations Research**, Moscow, Russia, October 15-19, 2013. Presented parts of Paper I.
7. An invited plenary talk at **The 3rd International Conference on Optimization and Numerical Analysis**, Muscat, Oman, January 6-9, 2014. Presented parts of Paper I.

Unconstrained Optimization

1 Introduction

The first part of this thesis gives an overview of optimization methods aimed at solving unconstrained optimization problem formulated as

$$\min_{x \in \mathbb{R}^n} f(x), \quad (2.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is assumed to be a twice continuously differentiable function. This overview is mainly based on materials from [15, 28, 49, 61].

1.1 Optimality conditions

Ideally, we want to find a *global minimizer* of f , that is, a point x^* such that

$$f(x^*) \leq f(x)$$

for all x where f is defined, i.e., $x \in \text{dom}(f)$. Unfortunately, for most of the problems it is not possible to know the overall shape of the function and to verify if a given point is a global minimizer. Typically, unconstrained optimization methods search for a *local* minimizer, where $f(x^*) \leq f(x)$ for all x in some neighborhood of x^* . We say, that x^* is a *strict* local minimizer, if there exists such a neighborhood, where $f(x^*) < f(x)$ for all $x \neq x^*$. If there is such a neighborhood, where x^* is the only local minimizer, it is an *isolated* local minimizer. Notice, that every isolated local minimizer is a strict local minimizer, but not vice versa.

To recognize a solution, we use the *first-order necessary* optimality conditions: if x^* is a local minimizer, then

$$\nabla f(x^*) = 0. \quad (2.2)$$

However, not every *stationary* point, where (2.2) is satisfied, is a local minimizer. A simple counter-example is $x^* = 0$ for $f(x) = x^3$.

Stronger conditions require the second-order information about the Hessian of the function in x^* , which is the matrix of second partial derivatives of f :

$$\nabla^2 f(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{\substack{i=1,\dots,n \\ j=1,\dots,n}}.$$

For their formulation, we introduce the following definitions.

Matrix A is called *positive semi-definite* ($A \geq 0$), if $d^T A d \geq 0$ for all $d \in \mathbb{R}^n$. It is called *positive definite* ($A > 0$), if $d^T A d > 0$ for all $d \neq 0$.

The *second-order necessary* optimality conditions state: if x^* is a local minimizer, then

$$\nabla f(x^*) = 0, \quad \nabla^2 f(x^*) \geq 0.$$

These conditions themselves do not ensure that a point is a local minimizer.

The *second-order sufficient* optimality conditions state: if x^* is such that

$$\nabla f(x^*) = 0, \quad \nabla^2 f(x^*) > 0,$$

then x^* is a strict local minimizer. Notice, that x^* can be a strict local minimizer while the sufficient conditions do not hold. As an example, consider $x^* = 0$ for $f(x) = x^4$.

We cannot generally say if a given point is a global minimizer, except some special cases. For instance, when f is a *convex* function, that is, if

$$\lambda x' + (1 - \lambda)x'' \in \text{dom}(f) \tag{2.3}$$

$$f(\lambda x' + (1 - \lambda)x'') \leq \lambda f(x') + (1 - \lambda)f(x'') \tag{2.4}$$

for any $x', x'' \in \text{dom}(f)$ and $\lambda \in [0, 1]$. Based on these properties, one can prove that any local minimizer of a convex function is global.

A convex function is *strictly* convex if a strict inequality holds in (2.4) for $\lambda \in (0, 1)$. A strictly convex function has a unique stationary point, which is its global minimizer.

Unfortunately, convexity is a very strong requirement and we usually deal with non-convex functions in practice. But many optimization methods are based on convex approximations of the objective function.

We further describe iterative *local* optimization methods aimed at finding a local minimizer of f , but the obtained solution could be just a stationary point. They require a starting point x_0 and generate a sequence of iterates $\{x_k\}$ to approximate this solution. Typically, each new iterate x_{k+1} is chosen to provide improvement in the objective function, i.e., $f(x_{k+1}) < f(x_k)$. Depending on the strategy for choosing x_{k+1} , there are two main approaches: line-search methods and trust-region methods. They both use a quadratic model of the objective

function, which is constructed using information about f from x_k and possibly from previous iterations:

$$f(x_k + s) \approx f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s \equiv q_k(s), \quad (2.5)$$

where $g_k = \nabla f(x_k)$ and B_k is either the true Hessian in x_k or its approximation. For simplicity, we will further denote $\nabla^2 f_k = \nabla^2 f(x_k)$.

1.2 Line-search methods

Line-search methods generate a search direction p_k by finding an unconstrained minimizer of a model function. This involves solving the system of linear equations

$$B_k p_k = -g_k. \quad (2.6)$$

The new iterate is computed as

$$x_{k+1} = x_k + \alpha_k p_k, \quad (2.7)$$

where the steplength α_k is determined by approximately minimizing f along the search direction:

$$\min_{\alpha > 0} f(x_k + \alpha p_k). \quad (2.8)$$

Finding the exact minimizer of (2.8), in general, is as difficult as to solve the original problem. Inexact line-search procedures typically calculate α_k such that in the new iterate the Wolfe conditions

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k g_k^T p_k, \quad (2.9a)$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 g_k^T p_k \quad (2.9b)$$

are satisfied with $0 < c_1 < c_2 < 1$ [69]. Here, the first inequality ensures sufficient reduction in the new iterate, and the second one prevents the steplength to be too short. Practical methods often use the strong Wolfe conditions (2.9a) and

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq -c_2 g_k^T p_k. \quad (2.10)$$

To formulate convergence properties of line-search methods, let us to introduce the following definition. Function $h : \mathcal{D} \rightarrow \mathbb{R}^m$ where $\mathcal{D} \subset \mathbb{R}^n$ is said to be *Lipschitz continuous* on some set $\mathcal{N} \subset \mathcal{D}$ if there exists a constant $L > 0$ such that

$$\|h(x') - h(x'')\|_2 \leq L \|x' - x''\|_2,$$

for all $x', x'' \in \mathcal{N}$. The constant L is called the *Lipschitz constant*.

Convergence of line-search methods depends both on the choice of the search direction and the steplength. Vector $p \in \mathbb{R}^n$ is called a *descent* direction, if

$$g_k^T p \leq 0.$$

This property is very important, because according to the Taylor's series, f locally decreases along any descent direction.

The following result establishes the convergence properties of inexact line-search methods.

Theorem 1 [49] *Consider any iteration of the form (2.7), where p_k is a descent direction and α_k satisfies the Wolfe conditions (2.9). Suppose that f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} \equiv \{x : f(x) \leq f(x_0)\}$, where x_0 is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on $\mathcal{N}$. Then*

$$\sum_{k \geq 0} \cos^2 \theta_k \|g_k\|_2^2 < \infty, \quad (2.11)$$

$$\text{where } \cos \theta_k = \frac{-g_k^T p_k}{\|g_k\|_2 \|p_k\|_2}.$$

The Zoutendijk condition (2.11) implies

$$\cos^2 \theta_k \|g_k\|_2 \rightarrow 0.$$

Therefore, if the search directions are never too orthogonal to the gradient, that is, there exists a scalar $\nu > 0$ such that

$$\cos \theta_k \geq \nu > 0, \quad \forall k, \quad (2.12)$$

then the iterations converge to a stationary point

$$\lim_{k \rightarrow \infty} g_k = 0.$$

To guarantee convergence to a stationary point, which is a local minimizer, additional assumptions are required.

Although many unconstrained optimization methods *globally* converge to a stationary point x^* from an arbitrary starting point, they differ in the rate of convergence. Below we provide quotient-based classification of the rate of convergence, starting from the slowest, where $\|\cdot\|$ denotes some vector norm.

1. Linear convergence: $\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq r$ for $r \in (0, 1)$ and all k sufficiently large.
2. Superlinear convergence: $\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$.
3. Quadratic convergence: $\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq M$ for $M > 0$ and all k sufficiently large.

Such rates of convergence often have only local character, when methods are already near the solution. We further review some of well-known line-search methods and their convergence properties.

The *steepest descent* method exploits only first-order information on f to generate a search direction, i.e., $B_k = I$ and

$$p_k = -\nabla f(x_k). \quad (2.13)$$

This is always a descent direction, which requires only one gradient to compute. It can easily be shown, that property (2.12) is satisfied and hence the steepest descent is globally convergent. But the rate of convergence is typically very slow and superseded by other methods. Even for a convex quadratic function with exact line-search this method converges linearly.

In the *Newton* method, $B_k = \nabla^2 f_k$ is the true Hessian and

$$p_k^N = -\nabla^2 f_k^{-1} g_k. \quad (2.14)$$

Under strict assumptions on f , the Newton method is theoretically the fastest unconstrained optimization method. Namely, if $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of a solution x^* , the Newton method converges quadratically to x^* from a sufficiently close starting point x_0 with the steplength $\alpha_k = 1$.

In practice, the Newton method may fail whenever $\nabla^2 f_k$ is not positive definite, because we can not guarantee, that the Newton direction exists, and even if it exists, that it is a descent direction.

There is a number of approaches developed to modify the Hessian and use a positive definite B_k in (2.6). This ensures that p_k is a descent direction. The simplest approach is to augment the Hessian

$$B_k = \nabla^2 f_k + \Delta B_k, \quad \Delta B_k > 0,$$

where typically $\Delta B_k = \gamma_k I$. The choice of γ_k depends on the smallest eigenvalue of $\nabla^2 f_k$:

$$\gamma_k > \max \{0, -\lambda_{\min}(\nabla^2 f_k)\}.$$

Practical implementations do not require expensive eigenvalue decomposition of the Hessian but are based on Gaussian elimination. Other approaches are based on modifications of the Cholesky factorization [26] or the symmetric indefinite factorization [13] of the Hessian.

Inexact Newton methods find search direction by approximately solving (2.6). They are efficient for large-scale problems, when n is large and finding an exact solution of (2.6) is computationally costly. We review some of these methods in section 4.1.

The main feature for most of the approaches presented above is the requirement to compute the Hessian, which is generally an expensive procedure, especially, for large-scale problems. Moreover, analytic representations of the second

derivatives are unavailable for many practical problem. One approach to avoid direct computation is to use finite differences (see, e.g., [49]). This would require additional costs for either computing n gradients or $O(n^2)$ function evaluations per iteration. Automatic differentiation [30] is an alternative approach proved to be successful for some of the cases.

Quasi-Newton methods represent another popular in practice approach, when B_k is an approximation to the true Hessian updated after each successful iteration using information about the recent change of the gradient. We review in more details some of these methods in section 3.

To summarize, line-search methods generate a search direction by solving the linear system (2.6). For their convergence, it is necessary that the search direction is a descent direction, which is only guaranteed when B_k is positive definite. To determine the steplength along the search direction, line-search procedures are often based on the Wolfe or the strong Wolfe conditions and require additional function and gradient evaluations per iteration.

In the next subsection, we introduce trust-region methods which provide an alternative approach to computing x_{k+1} . They do not require B_k to be positive definite, which allows to exploit negative curvature information and make more successful steps.

1.3 Trust-region methods

Trust-region methods at each iteration generate a trial step s_k by minimizing the model function inside the trust region, which is a ball of radius Δ_k

$$\Omega_k = \{s \in \mathbb{R}^n : \|s\| \leq \Delta_k\}.$$

Here $\|\cdot\|$ denotes a vector norm, typically, l_2 or l_∞ norms. We further assume the Euclidean norm in the trust region. The trust-region subproblem is formulated as follows:

$$\min_{s \in \Omega_k} f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s. \quad (2.15)$$

The new iterate is computed as

$$x_{k+1} = x_k + s_k.$$

To decide whether to accept this point, trust-region methods compute the ratio

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{q_k(s_k) - q_k(0)}. \quad (2.16)$$

Ideally, ρ_k should be close to 1, which indicates close approximation of the objective function by its model. If ρ_k is sufficiently positive, the trial step is accepted with possibly increased trust-region radius at the next iteration.

Otherwise, the trial step is rejected and $x_{k+1} = x_k$. In the later case, the trust-region radius Δ_k is decreased at the next iteration.

An advantage of trust-region methods is that they can naturally treat the case, when the Hessian or its approximation in the model function q is not positive definite. The trust-region constraint prevents from too long steps along negative curvature directions as opposed to the line-search methods, in which case there could be no solution to (2.8).

Trust-region methods are classified as nearly-exact and inexact methods, depending on how accurately the trust-region subproblem (2.15) is solved. In the case of the Euclidean norm, the exact solution s_k^* to (2.15) is characterized by the following optimality conditions [46]. There is a unique pair (s_k^*, σ_k^*) with $\sigma_k^* \geq 0$ such that:

$$\begin{aligned} (B_k + \sigma_k^* I) s_k^* &= -g_k, \\ \sigma_k^* (\|s_k^*\|_2 - \Delta_k) &= 0, \\ B_k + \sigma_k^* I &\geq 0. \end{aligned} \tag{2.17}$$

The Moré-Sorenson approach [46] seeks for the optimal pair (s^*, σ^*) that satisfies conditions (2.17). If the Newton step

$$s_k^N = -B_k^{-1} g_k$$

does not belong to the trust region, the solution to the trust-region subproblem is defined by the equation

$$\phi_k(\sigma) = 0, \tag{2.18}$$

where $\phi_k(\sigma) = 1/\Delta_k - 1/\|s\|$ and $s = s(\sigma)$ is the solution to the linear system

$$(B_k + \sigma I)s = -g_k. \tag{2.19}$$

Equation (2.18) is solved by Newton's root-finding algorithm [46]. At each iteration, the Cholesky factorization of $B_k + \sigma I$ is computed for solving the systems of linear equations (2.19). In practice, just a couple of iterations are required to be done for solving (2.18) to an appropriate accuracy.

For global convergence of trust-region methods, it is sufficient to find a solution to (2.15), such that the model decrease is at least a fixed fraction of that attained by the so-called Cauchy point [15], which we denote as p_k^C . This point is the minimizer of q_k along the steepest descent direction $-g_k$ subject to the trust-region constraint:

$$s_k^C = -\mu_k g_k, \quad \mu_k = \min \left\{ \frac{\|g_k\|_2^2}{g_k^T B_k g_k}, \frac{\Delta_k}{\|g_k\|_2} \right\}. \tag{2.20}$$

We say, that the trust-region subproblem is solved to a *sufficient accuracy*, if there exists a scalar $0 < c < 1$ such that the inequality

$$q_k(s_k) \leq -c \|g_k\|_2^2 \min \left(\frac{1}{\|B_k\|_2}, \frac{\Delta}{\|g_k\|_2} \right)$$

holds for all $k \geq 0$.

In Algorithm 1, we present a generic trust-region framework [15] in the form, close to our implementation in Paper I.

Algorithm 1 Trust-Region Algorithm

Require: $x_0 \in \mathbb{R}^n$, $\Delta_0 > 0$, $\epsilon > 0$, $0 \leq \tau_0 < \tau_1 < 1 < \tau_2$, $0 < \tau_3 < \tau_4 < 1$,
 Compute g_0 and B_0 .
for $k = 0, 1, 2, \dots$ **do**
     if $\|g_k\|_2 \leq \epsilon$ **then**
         return
     end if
     Find s_k that solves (2.15) to a sufficient accuracy.
     Compute the ratio ρ_k .
     if $\rho_k > \tau_0$ **then**
          $x_{k+1} = x_k + s_k$
         Compute g_{k+1} and B_{k+1} .
     else
          $x_{k+1} = x_k$
     end if
     if $\rho_k \geq \tau_1$ and $\|s_k\|_2 \geq 0.8\Delta_k$ **then**
          $\Delta_{k+1} = \tau_2\Delta_k$
     else
          $\Delta_{k+1} = \max(\tau_3\|s_k\|_2, \tau_4\Delta_k)$
     end if
end for

Suppose that Algorithm 1 generated a sequence $\{x_k\}$ such that B_k are *uniformly bounded above*, that is, there exists $c > 0$ such that

$$\|B_k\|_2 \leq c, \quad \forall k \leq 0. \quad (2.21)$$

If $\tau_0 = 0$, then the following result holds under certain assumptions on f :

$$\liminf_{k \rightarrow \infty} \|g_k\|_2 = 0.$$

A stronger result holds if $\tau_0 \in (0, \frac{1}{4}]$:

$$\lim_{k \rightarrow \infty} g_k = 0.$$

For proof, see Theorems 4.5 and 4.6 in [49].

Inexact trust-region methods generally require more iterations to converge, but they gain in finding an approximate solution to the trust-region subproblem at a lower computational cost than nearly-exact methods. In this section, we review some of those methods, based on linear combinations of the steepest descent

direction and the Newton direction. In section 4.2, we present the truncated CG method [60, 64], which is effective for solving large-scale problems. A large overview of trust-region methods is given in [15].

Dogleg approach [54, 55] finds the minimizer of the model function along a piece-wise linear path, which begins in $s = 0$, ends in the Newton step and has the Cauchy point as a knot. Since $q_k(s)$ and $\|s\|_2$ are monotonically decreasing and increasing, respectively, along the dogleg path, the minimizer of $q_k(s)$ on the feasible segment of the path is the endpoint of the segment. Thus,

$$s_k = \begin{cases} s_k^N, & \text{if } \|s_k^N\| \leq \Delta_k, \\ s_k^C + \theta_k(s_k^N - s_k^C), & \text{otherwise,} \end{cases} \quad (2.22)$$

where $\theta_k \in [0, 1)$ is such that $\|s_k\|_2 = \Delta_k$. In practice, the double-dogleg [18] method is usually more successful, because it generates a trial step which is closer to the Newton step.

An alternative approach is to search for the trust-region solution in the two-dimensional subspace spanned by s_k^C and s_k^N [11]. In this case, the trust-region subproblem (2.15) is augmented with an additional constraint

$$s \in \text{span} \{g_k, B_k^{-1}g_k\}.$$

Clearly, *two-dimensional subspace minimization* is an extension of the dogleg approach, since the dogleg path is feasible for the new constraint. This approach requires a single matrix factorization but often produces a similar reduction in the model function, as a nearly-exact trust-region solution, which requires 2-3 matrix factorizations [49].

Trust-region Newton methods take $B_k = \nabla^2 f_k$. Under certain assumptions on f , they converge superlinearly to a point, where the second-order sufficient optimality conditions are satisfied. Practical implementations of the presented above trust-region methods, both exact and inexact, take steps $s_k = s_k^N$ near the solution and hence converge quadratically.

In section 3 devoted to quasi-Newton methods, we describe some of their combinations with the trust-region framework. Paper I presents efficient combinations of the trust-region framework with limited memory quasi-Newton methods discussed in section 4.3.

We conclude this section with a summary in which line-search and trust-region methods are compared. Their success is problem dependent. Line-search procedures are often based on the strong Wolfe conditions (2.9a) and (2.10) that involve additional function and gradient evaluations. Line search methods require B_k to be positive definite, while trust-region methods can gain from exploiting information about possible negative curvature. Moreover, the latter methods compute gradients only when the trial step is accepted.

2 Conjugate gradient methods

In this section, we review conjugate gradient (CG) methods. The linear CG methods are widely used for solving large linear systems, while their nonlinear version is among the most popular methods for solving large-scale unconstrained optimization problems. Nonlinear CG have a number of advantages, where the low storage requirement and inexpensive algebraic costs are crucial. At each iteration, they require a few evaluations of function and its gradient without involving matrix operations, but have faster convergence, than the steepest descent method.

2.1 Linear conjugate gradient methods

Consider the problem of solving a linear system of equations

$$Ax = b, \quad (2.23)$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix and $b \in \mathbb{R}^n$. Problem (2.23) can be alternatively formulated as an unconstrained optimization problem of type (2.1):

$$\min_x \phi(x) = \frac{1}{2}x^T Ax - b^T x. \quad (2.24)$$

Indeed, this is a strictly convex quadratic problem, which has the unique global minimizer at a point, where the residual $r(x) = Ax - b$ in (2.23) is zero:

$$\nabla \phi(x) = Ax - b = 0$$

The CG method generates a sequence of so-called *conjugate* directions $p_0, p_1, \dots, p_l$ with respect to A , which satisfy

$$p_i^T A p_j = 0, \quad \forall i \neq j.$$

One can easily prove, that any set of conjugate directions is linearly independent.

Assume, for now, that we are given a set of n conjugate directions. Then the conjugate directions (CD) method solves (2.24) in at most n iterations of the form

$$x_{k+1} = x_k + \alpha_k p_k, \quad (2.25)$$

where

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} \quad (2.26)$$

is the exact minimizer of $\phi(x)$ along $x_k + \alpha p_k$ and $r_k = r(x_k)$.

Let x^* be the solution to (2.24). One can verify convergence of CD by showing that the exact representation of $x^* - x_0$ in the basis composed of the conjugate directions is as follow

$$x^* - x_0 = \alpha_0 p_0 + \alpha_1 p_1 + \cdots + \alpha_{n-1} p_{n-1}.$$

The CG method is a special case of CD. Denote $\mathcal{S}_k = \text{span}\{p_0, p_1, \dots, p_{k-1}\}$. The following properties of CD lead to the development of the CG method.

Consider a sequence $\{x_k\}$ generated by the CD method using (2.25) and (2.26). It immediately follows that

$$r_{k+1} = r_k + \alpha_k A p_k.$$

Moreover, one can derive

$$x_{k+1} = \arg \min_{x \in x_0 + \mathcal{S}_k} \phi(x), \quad (2.27)$$

$$r_{k+1} \perp \mathcal{S}_k. \quad (2.28)$$

Here, the first observation implies, that CD seeks for a minimizer of $\phi(x)$ in a sequence of expanding subspaces, and the second one implies, that the current residual is orthogonal to all previous search directions. These observations imply that the CG method formulated in Algorithm 2, converges to x^* in at most n steps. For details, see, e.g., [49].

Algorithm 2 CG Algorithm

Require: $x_0 \in \mathbb{R}^n$.

$r_0 = Ax_0 - b$, $p_0 = -r_0$, $k = 0$.

while $r_k \neq 0$ **do**

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k};$$

$$x_{k+1} = x_k + \alpha_k p_k;$$

$$r_{k+1} = Ax_{k+1} - b;$$

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k};$$

$$p_{k+1} = -r_{k+1} + \beta_k p_k;$$

$$k = k + 1;$$

end while

In the practical implementations of the CG method, the following formulas are used:

$$\alpha_k = -\frac{r_k^T r_k}{p_k^T A p_k} \quad (2.29)$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \quad (2.30)$$

The major computational burden in one CG iteration is related to the matrix-vector multiplication Ap_k at a cost of n^2 operations, which makes the total cost of solving (2.23) to be at most $n^3 + O(n^2)$. The CG method is recommended only for large-scale problems, otherwise it is usually outperformed by algorithms based on matrix factorizations, such as Gaussian elimination or the singular value decomposition (SVD). The advantage of the CG method is that it may converge in significantly fewer number of iterations than n .

The convergence of CG is based on distribution of eigenvalues of A . if A has only m distinct eigenvalues, CG converges to the solution in at most m iterations [49]. In the general case, when A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, we obtain

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x^*\|_A^2,$$

where $\|x\|_A^2 = x^T A x$. Thus, if the eigenvalues of A are well-clustered, say, there are m clusters, then CG produce a good approximation to the solution in m iterations. The CG method has the worst convergence when the eigenvalues are randomly distributed with $\lambda_1 \ll \lambda_n$ and A is poorly conditioned. The remedy is to use a preconditioning non-singular matrix C such that

$$C^{-T} A C^{-1} \approx I$$

and to solve instead of the original linear system (2.23) an equivalent linear system

$$(C^{-T} A C^{-1}) \hat{x} = C^{-T} b$$

in the new variables

$$\hat{x} = Cx.$$

Practical preconditioned CG methods provide a fast rate of convergence at the additional cost per iteration which is similar to the cost of a matrix-vector product.

2.2 Nonlinear conjugate gradient methods

The nonlinear CG method is a line-search method aimed at solving (2.1). It is an extension of Algorithm 2 with r_k being replaced by the function gradient g_k . In this method, each new iterate is computed as

$$x_{k+1} = x_k + \alpha_k p_k,$$

where the new search direction p_k depends only on the previous one:

$$p_{k+1} = -g_k + \beta_k p_k, \quad p_0 = -g_0.$$

Unlike the linear CG, the steplength α_k is found by an inexact line-search procedure. The strong Wolfe conditions (2.9a) and (2.10) ensure that p_{k+1} is a

descent direction with the initial search direction $p_0 = -g_0$. Different choice of β_k leads to variants of the nonlinear CG. The first version was proposed by Fletcher and Reeves [25]

$$\beta_{k+1}^{\text{FR}} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k},$$

which is the direct analog of the same parameter (2.30) in the linear CG. In fact, for a convex quadratic $f(x)$, the Fletcher-Reeves algorithm is exactly the linear CG algorithm, described above.

Other classic variants of the CG method include the formulas by Polak and Ribière [53]

$$\beta_{k+1}^{\text{PR}} = \frac{g_{k+1}^T y_k}{g_k^T g_k}$$

and by Hestenes and Steifel in their original paper on linear CG method [36]

$$\beta_{k+1}^{\text{HS}} = \frac{g_{k+1}^T y_k}{y_k^T p_k},$$

where $y_k = g_{k+1} - g_k$.

Competitive modern variants include the formulas by Dai and Yuan [16]

$$\beta_{k+1}^{\text{DY}} = \frac{\|g_{k+1}\|_2^2}{y_k^T p_k}$$

and by Hager and Zhang [32]

$$\beta_{k+1}^{\text{HZ}} = \left(y_k - 2p_k \frac{\|y_k\|^2}{y_k^T p_k} \right)^T \frac{g_{k+1}}{y_k^T p_k}.$$

The last version β_{k+1}^{HZ} is arguably the most successful nonlinear CG algorithm in practice, according to the reported numerical experiments. In this algorithm, the authors use a special line-search procedure, which is based on approximate Wolfe conditions. A large survey of CG methods is given, e.g., in [33].

Under standard assumptions in Theorem 1 and the Zoutendijk condition (2.11), the CG methods converge globally. For faster convergence, CG methods typically use restarts, whenever two consecutive search directions loose orthogonality. After the restart, it is typical to choose the steepest descent direction as an initial direction for the CG method. Recently, limited memory techniques have been proposed in [34] to generate a new search direction and restore orthogonality of the CG directions, which allows to avoid restarts.

Nonlinear CG methods are very attractive for solving large-scale problems, because at each iteration they do not require any operations involving matrices to compute the search direction, but to store few vectors of length n and to perform computationally inexpensive operations of order $O(n)$.

3 Quasi-Newton methods

Quasi-Newton methods iteratively solve (2.1) by using in (2.6) a Hessian approximation B_k , which is updated after each successful iteration exploiting the recent change of the gradient. Quasi-Newton updates for Hessian approximation have a general form

$$B_{k+1} = B_k + \Delta B_k,$$

where the choice of ΔB_k is required to satisfy the *secant equation*

$$B_{k+1} s_k = y_k \quad (2.31)$$

for $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$ and the symmetry property

$$B_{k+1}^T = B_{k+1}. \quad (2.32)$$

The choice of ΔB_k that satisfies (2.31) and (2.32) is not unique. But under assumption that ΔB_k is a symmetric rank-one matrix, there is a unique solution, which yields the *symmetric-rank-one* (SR1) update

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \quad (2.33)$$

If ΔB_k is a rank-two matrix, the solution is not anymore unique, which leads to a variety of quasi-Newton updates. The *BFGS* update, named after Broyden, Fletcher, Goldfarb and Shanno, is generally considered to be one of the most successful quasi-Newton updates in practice:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (2.34)$$

We mention that both SR1 and BFGS belong to the Broyden class of quasi-Newton updates

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \theta w_k w_k^T, \quad (2.35)$$

for the appropriate choice of θ , where

$$w_k = (s_k^T B_k s_k)^{1/2} \left(\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right).$$

For overview of other quasi-Newton updates, we refer reader, e.g., to [1, 19].

Both SR1 and BFGS have formulas for computing inverse Hessian approximations $H_k = B_k^{-1}$, namely, for the SR1 update

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \quad (2.36)$$

and for the BFGS update

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \quad (2.37)$$

where

$$\rho_k = \frac{1}{y_k^T s_k}. \quad (2.38)$$

This allows for efficient computation of the quasi-Newton direction in the line-search framework at the cost of $O(n^2)$ operations.

For the BFGS update, the secant equations (2.31) combined with Wolfe conditions (2.9a) and (2.9b) (or strong Wolfe conditions) ensure that B_{k+1} is positive definite whenever B_k is positive definite, because the curvature condition

$$s_k^T y_k > 0$$

is satisfied by Wolfe conditions [49]. It is popular in practice to choose the initial approximation as

$$H_0 = \gamma_k I,$$

with

$$\gamma_k = \frac{y_k^T s_k}{y_k^T y_k}. \quad (2.39)$$

Moreover, it was shown that the BFGS updates have self-correcting properties, that allows to maintain a good approximation of the function curvature and be uniformly bounded above (2.21).

The SR1 update is not guaranteed to generate a positive definite Hessian approximation and the method may fail. For this reason, SR1 updates are often combined with the trust-region framework. In this case, even if the trial step is rejected, it is recommended to update the Hessian approximation, because this prevents from searching along the same direction at the next iterations. Practical implementations of SR1 method skip update, whenever the denominator is too small

$$|s_k^T (y_k - B_k s_k)| < \varepsilon \|s_k\|_2 \|y_k - B_k s_k\|_2$$

with $\varepsilon > 0$ sufficiently small.

We now briefly discuss convergence properties of quasi-Newton methods. Consider first a strongly convex quadratic function. Then any method from the Broyden class with exact line-search procedure converges to the solution at most in n iterations. Moreover, if $B_0 = I$, then it produces the same iterations as the CG method (Theorem 6.4, [49]).

In case of a general function, the BFGS method with line-search based on the Wolfe conditions locally converges superlinearly under some assumptions. For details, see Theorem 6.6 in [49].

The SR1 method has different convergence properties than BFGS. In case of a strongly convex quadratic function with inexact line search, it will converge

in at most $n + 1$ iterations if the generated steps are linearly independent [24]. Moreover, B_{n+1} is identical to the true Hessian. This is not generally true for other quasi-Newton methods, which makes attractive the SR1 update.

For a general function, convergence properties for the SR1 method have been established under stronger assumptions, than in case of the BFGS method. In [14], superlinear convergence is established if the generated steps are uniformly linearly independent. If to disregard uniform linear independence, $(n + 1)$ -superlinear convergence

$$\lim \frac{\|x_{k+n+2} - x^*\|_2}{\|x_k - x^*\|_2} = 0$$

is established in [38], when B_k are assumed to be positive definite and uniformly bounded. In practice, the SR1 update tends to approximate the true Hessian near the solution better than other quasi-Newton updates.

Lastly, we discuss multipoint symmetric secant methods proposed in [5, 6, 7, 8]. In these methods, the Hessian is approximated by virtue of the following rank-two updating formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) c_k^T + c_k (y_k - B_k s_k)^T}{s_k^T c_k} - \frac{(y_k - B_k s_k)^T s_k c_k c_k^T}{(s_k^T c_k)^2}, \quad (2.40)$$

where $c_k \in \mathbb{R}^n$ is such that

$$\begin{aligned} c_k^T s_i &= 0, & \text{for some } i < k, \\ c_k^T s_k &\neq 0. \end{aligned}$$

They extend the classical multipoint secant methods [51] and exploit the symmetry of the Hessian matrix in a natural way. Like SR1, they are able to accumulate important negative curvature information, which is not inherent in BFGS.

To summarize, quasi-Newton methods require a low-rank update of the approximation to the Hessian or its inverse at the cost of $O(n^2)$ operations and the cost of the same order to compute the search direction. Under certain conditions, they converge superlinearly. In the trust-region framework, SR1 and multipoint symmetric secant approximations update have advantage in exploring negative curvature information.

4 Large-scale optimization methods

In this section, we overview methods for solving (2.1), when number of decision variables n is large and problem is considered to be large-scale [28]. In section 4.1, we have already presented nonlinear CG methods and described their advantages in solving large-scale problems. These methods use only first-order information about f and require $O(n)$ operations per iteration.

The Newton-like methods exploit the second-order information and generally converge faster, but their direct implementation requires at least $O(n^2)$ operations per iteration, which is too costly, when n is large. We further describe inexact Newton methods where combinations with CG allow to decrease complexity of one iteration.

4.1 Line-search inexact Newton methods

In the line-search Newton methods, the main complexity is to compute the Newton direction by solving the linear system

$$\nabla^2 f_k p_k = -g_k. \quad (2.41)$$

The *Newton-CG* method [17], or truncated Newton method, approximately solves this system with the use of the linear CG method. It terminates solving (2.41) when

$$\|\nabla^2 f_k p_k + g_k\|_2 \leq \eta_k \|g_k\|_2,$$

with $0 < \eta_k < 1$ and $\eta_k \rightarrow 0$ if $g_k \rightarrow 0$.

At iteration k of the line-search method, CG generates conjugate directions $\{d_{k,j}\}$ with steplength along each direction determined by formula

$$\alpha_{k,j} = \frac{r_j^T r_j}{d_{k,j}^T \nabla^2 f_k d_{k,j}}.$$

Here the term $d_{k,j}^T \nabla^2 f_k d_{k,j}$ is computed with the use of finite differences or automatic differentiation, which allows to avoid calculation of the true Hessian. For instance, formula

$$\nabla^2 f_k d \approx \frac{\nabla f(x_k + hd) - g_k}{h}$$

gives $O(h)$ approximation error at the cost of one function evaluation.

If $d_{k,j}^T \nabla^2 f_k d_{k,j} < 0$, this test indicates that the Hessian is not positive definite, and the Newton-CG iteration is terminated by taking, e.g., the steepest descent direction.

Under standard assumptions and for some choice of η_k , the Newton-CG method converges superlinearly, e.g., $\eta_k = \min \left\{ 0.5, \sqrt{\|g_k\|_2} \right\}$ [49]. In practice, a well-preconditioned CG method requires a few iterations for this.

4.2 Trust-region inexact Newton methods

The early development of trust-region methods for large-scale problems is associated with the *truncated CG* method [60, 64]. It finds an approximate solution

to the trust-region subproblem by applying CG for solving the linear system (2.41). The idea of this method is based on two observations about the generated iterations $\{d_{k,j}\}$: the model function is monotonically decreasing

$$q_k(d_{k,j+1}) < q_k(d_{k,j}),$$

while the length of the step is growing

$$\|d_{k,j+1}\|_2 > \|d_{k,j}\|_2.$$

Eventually, either the unconstrained minimizer is found and the trial step is the Newton step, or $\|d_{k,j}\|_2 > \Delta_k$ for some j . In the later case, the trial step is laying on the boundary of the trust region and computed as

$$s_k = \frac{\Delta_k}{\|d_{k,j}\|_2} d_{k,j}.$$

All matrix-vector products involving the Hessian are performed in Hessian-free manner, as in case of the Newton-CG method.

The truncated CG method terminates whenever it finds a negative curvature direction. However, this may lead to insignificant changes in the model. More effective in treating negative curvature information is a similar trust-region approach, where the Lanczos method (see, e.g., [27]) is applied for solving (2.6) instead of CG. Practical implementations of both CG and Lanczos methods require preconditioning to accelerate the convergence.

Standard methods for finding nearly-exact solutions for the trust-region subproblem generally require $O(n^3)$ operations, which makes them impractical for large-scale problems. However, there were proposed reformulations of the trust-region subproblem as a parameterized eigenvalue problem [31, 56, 57, 59], which allows for finding a nearly-exact trust-region solution in the matrix-free manner.

Quasi-Newton updates typically generate dense Hessian approximations and matrix-vector multiplications require $O(n^2)$ operations. In the following section, we review limited memory quasi-Newton updates which allow for efficient computation of matrix-vector products without explicitly computing and storing the full matrix.

4.3 Limited memory quasi-Newton updates

Limited memory versions of the quasi-Newton update require to store only a limited number, $m \ll n$, of n -dimensional vectors, and the Hessian, or its inverse, is implicitly approximated by a low-rank update of a diagonal matrix. This allows for arranging efficient matrix-vector multiplications involving B_k and H_k .

So far, the most successful implementations of limited memory methods were associated with line search. Nowadays, the most popular limited memory line-search methods are based on the BFGS-update [49]. For this reason, we describe below the limited-memory BFGS update (L-BFGS). For more extensive overview of limited memory quasi-Newton updates, we refer reader to [41].

Consider the BFGS update for inverse Hessian approximation (2.37). In the L-BFGS update, H_k is obtained from an initial approximation H_0^k by applying (2.37) to the sequence of at most m pairs $\{s_i, y_i\}$. The following representation of the L-BFGS update holds:

$$H_k = V_{k-m_k, k}^T H_0^k V_{k-m_k, k} + \sum_{i=k-m_k}^{k-1} \rho_i V_{i+1, k}^T s_i s_i^T V_{i+1, k},$$

where $m_k = \min\{k, m\}$, ρ_i is defined by (2.38) and

$$V_{i, k} = \prod_{j=i}^k (I - \rho_j y_j s_j^T).$$

This representation allows to efficiently compute the quasi-Newton step $-H_k g_k$ at $4mn$ multiplications by a two-loop recursion procedure [48], formulated in Algorithm 3.

Algorithm 3 L-BFGS two-loop recursion

```

 $q = g_k$ 
for  $i = k - 1$  to  $k - m_k$  do
     $\alpha_i = \rho_i s_i^T q$ ;
     $q = q - \alpha_i y_i$ ;
end for
 $r = -H_0^k q$ ;
for  $i = k - m_k$  to  $k - 1$  do
     $\beta = \rho_i y_i^T r$ ;
     $r = r + s_i(\beta - \alpha_i)$ ;
end for
return  $r = -H_k g_k$ 

```

In practice, the initial inverse Hessian approximation has the form $H_0^k = \gamma_k I$ and the parameter γ_k is usually updated on the base of the most recent curvature information by the formula

$$\gamma_k = \frac{s_k^T y_k}{y_k^T y_k}. \quad (2.42)$$

In view of Paper I, we demonstrate further a compact representation of B_k , which is available for the most of the limited-memory quasi-Newton updates

$$B_k = \delta_k I + V_k W_k V_k^T, \quad (2.43)$$

where $W_k \in \mathbb{R}^{\bar{m}_k \times \bar{m}_k}$ is a symmetric matrix and $V_k \in \mathbb{R}^{\bar{m}_k \times \bar{m}_k}$ with $\bar{m}_k \ll n$ depending individually on the update. Such representation was originally derived for the BFGS and SR1 updates in [10].

Thus, the L-BFGS update can be represented as

$$B_k = \delta_k I - [S_k \ Y_k] \begin{bmatrix} S_k^T S_k / \delta & L_k / \delta_k \\ L_k^T / \delta_k & -E_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^T \\ Y_k^T \end{bmatrix}. \quad (2.44)$$

In terms of (2.43), $\bar{m}_k = 2m_k$ and $V_k = [S_k \ Y_k]$ consists of the stored couples $\{s_i, y_i\}$

$$S_k = [s_{k-m_k}, s_{k-m_k+1}, \dots, s_{k-1}], \quad Y_k = [y_{k-m_k}, y_{k-m_k+1}, \dots, y_{k-1}].$$

The matrix W_k is the inverse of an $2m \times 2m$ -matrix, which contains a strictly lower triangular part of the matrix $S_k^T Y_k$, denoted in (2.44) by L_k , and the main diagonal of $S_k^T Y_k$, denoted by E_k , i.e.,

$$\begin{aligned} (L_k)_{ij} &= \begin{cases} (s_{k-m_k-1+i})^T y_{k-m_k-1+j}, & \text{if } i > j, \\ 0, & \text{otherwise,} \end{cases} \\ (E_k)_{ij} &= \begin{cases} (s_{k-m_k-1+i})^T y_{k-m_k-1+i}, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Finally, $\delta_k = 1/\gamma_k$. A similar to (2.44) compact representation is available for the inverse Hessian approximation by the L-BFGS update. It has the same complexity of computing the quasi-Newton direction as the aforementioned two-loop recursion procedure.

We also provide here compact representations of the SR1 update and multipoint symmetric secant approximations, which we refer to in Paper I. For SR1, the following compact representation holds:

$$B_k = \delta_k I + [S_k \ Y_k] \begin{bmatrix} -\delta_k I \\ I \end{bmatrix} (E_k + L_k + L_k^T - \delta_k S_k^T S_k)^{-1} \begin{bmatrix} -\delta_k I \\ I \end{bmatrix}^T \begin{bmatrix} S_k^T \\ Y_k^T \end{bmatrix}. \quad (2.45)$$

This expression differs from the original one in [10], because it allows to exploit techniques proposed in Paper I.

For multipoint symmetric secant approximations, the compact representation has been derived in [8]:

$$B_k = \delta_k I + [\bar{S}_k \ \bar{Y}_k] \begin{bmatrix} -M_k \text{sym}(\bar{Y}_k^T \bar{S}_k) M_k - \delta_k M_k & M_k \\ M_k & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{S}_k^T \\ \bar{Y}_k^T \end{bmatrix}, \quad (2.46)$$

where $M_k = (\bar{S}_k^T \bar{S}_k)^{-1}$,

$$\bar{S}_k = [s_{k-1}, s_{k-2}, \dots, s_{k-m_k}], \quad \bar{Y}_k = [y_{k-1}, y_{k-2}, \dots, y_{k-m_k}].$$

Here, we use notation

$$\text{sym}(A) = \begin{cases} A_{ij}, & i \geq j, \\ A_{ji}, & \text{otherwise,} \end{cases}$$

defined for any square matrix A .

As it was mentioned above, there are problems where trust-region methods are expected to be more successful than line-search methods. Unfortunately, any straightforward embedding of limited memory quasi-Newton techniques in the trust-region framework deteriorates the efficiency of the former approach. The existing refined limited memory trust-region methods [9, 21, 22, 37] typically use the compact representations of limited-memory BFGS updates (2.37) for approximating the Hessian and the Euclidean norm for defining the trust region. In Paper I, we review these methods in details and present new limited memory trust-region algorithms which are competitive with line-search versions of the L-BFGS method.

5 Nonlinear least-squares problems

This section contains an overview of methods designed for solving least-squares problems

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \|r(x)\|^2, \quad (2.47)$$

where $r(x) = (r^1(x), \dots, r^m(x))^T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $m > n$.

The derivatives of the objective function in (2.47) are

$$\nabla f(x) = J(x)^T r(x), \quad (2.48)$$

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^m \nabla^2 r^i(x) r^i(x), \quad (2.49)$$

where $J(x) \in \mathbb{R}^{m \times n}$ is the Jacobian of $r(x)$ defined as

$$J(x) = \left[\frac{\partial r^i}{\partial x_j} \right]_{\substack{i=1, \dots, m \\ j=1, \dots, n}}.$$

Methods for solving nonlinear least-squares problems are based on observation that often a sufficiently good approximation to the Hessian is obtained by the first term in (2.49). The *Gauss-Newton* method is a modification of the standard Newton method by substituting $\nabla^2 f_k$ with $J_k^T J_k$ in (2.14), where $J_k = J(x_k)$. Denote also $r_k = r(x_k)$, then the Gauss-Newton search direction p_k^{GN} is computed from the normal equations

$$J_k^T J_k p_k^{GN} = -J_k^T r_k. \quad (2.50)$$

This system can be solved, e.g., by the linear CG method without computing explicitly $J_k^T J_k$.

Alternatively, the Gauss-Newton direction can be computed by solving the linear least-squares problem

$$\min_p \frac{1}{2} \|J_k p + r_k\|_2^2.$$

This can be done using QR or SVD decompositions of J_k and does not require to compute $J_k^T J_k$.

Convergence analysis of the Gauss-Newton method is based on the assumption that the Jacobians $J(x)$ are uniformly of full rank, i.e., there exists a scalar $\beta > 0$ such that

$$\|J(x)y\|_2 \geq \beta \|y\|_2, \quad \forall y \in \mathcal{N}, \quad (2.51)$$

where $\mathcal{N} \subset \mathbb{R}^n$ is determined by the starting point x_0 . Then the following results holds under certain additional assumptions (see Theorem 10.1 in [49]):

$$\lim_{k \rightarrow \infty} J_k^T r_k = 0.$$

The *Levenberg-Marquard* method is a modification of the trust-region version of the Newton method. It is also based on replacing $\nabla^2 f_k$ with $J_k^T J_k$, which leads to the trust-region subproblem

$$\min_{\|s\|_2 \leq \Delta_k} \frac{1}{2} \|J_k s + r_k\|_2^2. \quad (2.52)$$

An alternative approach is based on quasi-Newton approximations that exploit the special structure of the Hessian (2.49). For further details about these two approaches, we refer reader to [49].

In Paper II, we consider a special case of least-squares problems, called multi-linear least-squares (MLLS) problem. It originates from applied problems like design of filter networks. We present the MLLS problem in more details in section 3 of chapter 3.

6 Cardinality-constrained linear least-squares problem

In this section, we review some of sparse optimization methods aimed for solving a linear least-squares problem subject to a cardinality constraint

$$\min_x \|Ax - b\|^2 \quad \text{s.t.} \quad \|x\|_0 \leq K, \quad (2.53)$$

where $b \in \mathbb{R}^m$, $A = [a_1 \dots a_n] \in \mathbb{R}^{m \times n}$, $0 < K \leq n < m$ and $\|x\|_0$ denotes the number of non-zero components of $x \in \mathbb{R}^n$. Without loss of generality, the columns of A are of unit Euclidean length:

$$\|a_i\| = 1, \quad i = 1, \dots, n.$$

The combinatorial problem (2.53) is known to be NP-hard for general A and b [47]. We deal with this problem in Paper III and the efficiency of our algorithms depends on how efficiently it is solved.

6.1 Greedy pursuit methods

Greedy pursuit methods find an approximate solution to (2.53) by iteratively modifying one or few of its elements that yield the best improvement. In our numerical experiments related to design of filter networks, the most successful methods were the orthogonal matching pursuit (OMP) and the orthogonal least-squares (OLS) pursuit. To describe these methods, we need the following notations.

For any set $\Omega_k \subset \{1, \dots, n\}$ of cardinality k , let A_{Ω_k} denote a matrix composed of columns a_i of A , such that $i \in \Omega_k$, and $\bar{\Omega}_k$ is the complement set of Ω_k to Ω_n . By analogy, $x(\Omega_k)$ denotes a vector in $\mathbb{R}^k$ composed of elements x_i of vector x , such that $i \in \Omega_k$. Let also $(A_{\Omega_k})^\dagger \in \mathbb{R}^{n \times m}$ denote a pseudo-inverse matrix of A_{Ω_k} with the property that

$$(A_{\Omega_k})^\dagger A_{\Omega_k} = I.$$

Then the problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2 \quad \text{s.t.} \quad x(\bar{\Omega}_k) = 0, \quad (2.54)$$

is equivalent to the linear least-squares problem

$$\min_{x^k \in \mathbb{R}^k} \|A_{\Omega_k} x^k - b\|^2.$$

This allows to present the optimal solution to (2.54) in the form

$$x(\Omega_k) = (A_{\Omega_k})^\dagger b, \quad x(\bar{\Omega}_k) = 0.$$

Using these notations, the greedy pursuit methods OLS and OMP are presented in Algorithms 4 and 5, respectively. At iteration k , they extend the set of already selected columns Ω_{k-1} of matrix A by a new column i_k selected from the remaining columns using a greedy principle.

The selection criterion differs for OLS and OMP. In OLS, the new column together with those already selected gives the best least-squares approximation to b :

$$i_k = \arg \min_{i \in \Omega_{k-1}} \varepsilon_i, \quad \varepsilon_i = \|b - A_{\Omega_{k-1} \cup i} (A_{\Omega_{k-1} \cup i})^\dagger b\|^2. \quad (2.55)$$

In OMP, the new column has the largest correlation to the current residual

$$i_k = \arg \max_{i \in \Omega_{k-1}} r_{k-1}^T a_i, \quad r_{k-1} = b - A_{\Omega_{k-1}}(A_{\Omega_{k-1}})^\dagger b. \quad (2.56)$$

Algorithm 4 OLS

```

 $\Omega_0 = \emptyset, \bar{\Omega}_0 = \{1, \dots, n\};$ 
for  $k = 1, \dots, K$  do
  for  $i \in \bar{\Omega}_k$  do
     $\varepsilon_i = \|b - A_{\Omega_{k-1} \cup \{i\}}(A_{\Omega_{k-1} \cup \{i\}})^\dagger b\|^2;$ 
  end for
   $i_k = \arg \min_{i \in \bar{\Omega}_{k-1}} \varepsilon_i;$ 
   $\Omega_k = \Omega_{k-1} \cup \{i_k\};$ 
   $\bar{\Omega}_k = \bar{\Omega}_{k-1} \setminus \{i_k\};$ 
end for
return  $x$  with components  $x(\Omega_K) = (A_{\Omega_K})^\dagger b$  and  $x(\bar{\Omega}_K) = 0$ .
```

Algorithm 5 OMP

```

Ensure:  $\|a_i\| = 1, \quad i = 1, \dots, n.$ 
 $\Omega_0 = \emptyset, \bar{\Omega}_0 = \{1, \dots, n\}, r_0 = b;$ 
for  $k = 1, \dots, K$  do
   $i_k = \arg \max_{i \in \bar{\Omega}_{k-1}} r_{k-1}^T a_i;$ 
   $\Omega_k = \Omega_{k-1} \cup \{i_k\};$ 
   $\bar{\Omega}_k = \bar{\Omega}_{k-1} \setminus \{i_k\};$ 
   $r_k = b - A_{\Omega_k}(A_{\Omega_k})^\dagger b;$ 
end for
return  $x$  with components  $x(\Omega_K) = (A_{\Omega_K})^\dagger b$  and  $x(\bar{\Omega}_K) = 0$ .
```

Practical implementations of OLS and OMP maintain QR decomposition of A_{Ω_k} . This simplifies the selection procedure that requires computation of the current residual r_k for OMP and its norm for OLS.

Indeed, let $A_{\Omega_{k-1}} = Q_{k-1}R_{k-1}$, where $Q_{k-1} \in \mathbb{R}^{m \times (k-1)}$ is a matrix with orthonormal columns and $R_{k-1} \in \mathbb{R}^{(k-1) \times (k-1)}$ is an upper-triangular matrix. Then the selected column a_{i_k} at iteration k is split into two vectors

$$a_{i_k} = Q_{k-1}Q_{k-1}^T a_{i_k} + (I - Q_{k-1}Q_{k-1}^T) a_{i_k} = Q_{k-1}z_k + a_{i_k}^\perp, \quad (2.57)$$

where $z_k = Q_{k-1}a_{i_k} \in \mathbb{R}^{k-1}$. Denote the normalized orthogonal component

$$\bar{a}_{i_k}^\perp = a_{i_k}^\perp / \|a_{i_k}^\perp\|.$$

Then the update of the QR decomposition has the form

$$\begin{aligned} Q_k &= [Q_{k-1} \quad \bar{a}_{i_k}^\perp], \\ R_k &= \begin{bmatrix} R_{k-1} & z_k \\ 0 & \|a_{i_k}^\perp\| \end{bmatrix}. \end{aligned}$$

Observing that

$$A_{\Omega_{k-1}}(A_{\Omega_{k-1}})^\dagger = Q_{k-1}Q_{k-1}^T,$$

the new residual r_k can be represented as

$$r_k = (I - Q_k Q_k^T)b = (I - Q_{k-1} Q_{k-1}^T)b - \bar{a}_{i_k}^\perp (\bar{a}_{i_k}^\perp)^T b. \quad (2.58)$$

Here matrix $(I - Q_k Q_k^T)$ is a projection matrix, therefore equality (2.58) implies

$$\|r_k\|^2 = b^T (I - Q_{k-1} Q_{k-1}^T) b - (b^T \bar{a}_{i_k}^\perp)^2 = \|r_{k-1}\|^2 - (b^T \bar{a}_{i_k}^\perp)^2. \quad (2.59)$$

Using representation of a_{i_k} (2.57), the second term can be computed as

$$b^T \bar{a}_{i_k}^\perp = b^T (I - Q_{k-1} Q_{k-1}^T) a_{i_k} / \|a_{i_k}^\perp\| = (r_{k-1}^T a_{i_k}) / \|a_{i_k}^\perp\|. \quad (2.60)$$

Notice, that r_{k-1} is orthogonal to the selected columns $A_{\Omega_{k-1}}$, therefore

$$r_{k-1}^T a_{i_k} = r_{k-1}^T a_{i_k}^\perp.$$

This expression combined with (2.59) and (2.60) yields

$$\|r_k\|^2 = \|r_{k-1}\|^2 - (r_{k-1}^T \bar{a}_{i_k}^\perp)^2. \quad (2.61)$$

Formula (2.61) for updating the norm of the residual provides an equivalent selection criterion to (2.55)

$$i_k = \arg \max_{i \in \Omega_{k-1}} \bar{\phi}_i, \quad \bar{\phi}_i = r_{k-1}^T \bar{a}_i^\perp.$$

For comparison, the selection criterion for OMP can be rewritten as

$$i_k = \arg \max_{i \in \Omega_{k-1}} \phi_i, \quad \phi_i = r_{k-1}^T a_i^\perp.$$

This makes evident the difference between OMP and OLS. If OMP measures correlation of the residual to the orthogonal component of a_i , OLS measures correlation to the normalized orthogonal component, and they may select different columns.

We formulate a version of practical implementations of OLS and OMP as Algorithms 6 and 7, respectively. It is evident that OLS is more expensive, because at each iteration it requires to reorthogonalize the non-selected columns $A_{\bar{\Omega}_k}$ to those already selected.

We also note that when K columns are chosen, the corresponding linear least-squares problem is easy to compute using the QR decomposition. For this reason, a vector $y_{k-1} = Q_{k-1}^T b \in \mathbb{R}^{k-1}$ is augmented by $\bar{\phi}_{i_k}$ at each iteration and the final solution to (2.53) is computed as

$$x(\Omega_K) = R_K^{-1} y_K, \quad x(\bar{\Omega}_k) = 0.$$

Algorithm 6 OLS, practical form

Ensure: $\|a_i\| = 1, \quad i = 1, \dots, n.$
 $\Omega_0 = \emptyset, \bar{\Omega}_0 = \{1, \dots, n\}, r_0 = b;$
for $k = 1, \dots, K$ **do**
    for $i \in \bar{\Omega}_{k-1}$ **do**
         $a_i^\perp = a_i - \frac{a_i^T a_i^\perp}{\|a_i^\perp\|^2} a_i^\perp;$
         $\bar{\phi}_i = \frac{r_{k-1}^T a_i^\perp}{\|a_i^\perp\|};$
    end for
     $i_k = \arg \max_{i \in \bar{\Omega}_{k-1}} \bar{\phi}_i;$
     $\Omega_k = \Omega_{k-1} \cup \{i_k\};$
     $\bar{\Omega}_k = \bar{\Omega}_{k-1} \setminus \{i_k\};$
     $R_k = \begin{bmatrix} R_{k-1} & Q_{k-1}^T a_{i_k} \\ 0 & \|a_{i_k}^\perp\| \end{bmatrix};$
     $Q_k = \begin{bmatrix} Q_{k-1} & \frac{a_{i_k}^\perp}{\|a_{i_k}^\perp\|} \end{bmatrix};$
     $y_k = [y_{k-1}, \bar{\phi}_{i_k}];$
     $r_k = r_{k-1} - \frac{\bar{\phi}_{i_k}}{\|a_{i_k}^\perp\|} a_{i_k}^\perp;$
end for
return x with components $x(\Omega_K) = R_K^{-1} y_K$ and $x(\bar{\Omega}_K) = 0.$

Closely related to greedy pursuits are *iterative hard thresholding* methods, where the basic algorithm [3] starts from $x^0 = 0$ and the new iterate has the form

$$x^{k+1} = \lfloor x^k + A^\dagger A(b - Ax^k) \rfloor_K,$$

that is, the K largest by magnitude components are preserved in x_{k+1} after taking the Newton-type step.

6.2 Convex relaxation methods

The so called l_0 -norm $\|\cdot\|_0$ is a non-convex function. An alternative approach is based on convex relaxations of this norm, typically, by replacing it with l_1 -norm. This approach is effective in *compressed sensing* for recovering a sparse solution to an underdetermined system of linear equations, where $m \ll N$. Compressed sensing appears in many applications including medical imaging and there have been developed a large number of optimization methods for solving relaxed versions of the cardinality-constrained linear least-squares problem (2.53).

For instance, the least absolute shrinkage and selection operator problem with

---

Algorithm 7 OMP, practical form

Ensure: $\|a_i\| = 1, \quad i = 1, \dots, n.$
 $\Omega_0 = \emptyset, \bar{\Omega}_0 = \{1, \dots, n\}, r_0 = b;$
for $k = 1, \dots, K$ **do**
    for $i \in \bar{\Omega}_{k-1}$ **do**
         $\phi_i = r_{k-1}^T a_i;$
    end for
     $i_k = \arg \max_{i \in \bar{\Omega}_{k-1}} \phi_i;$
     $\Omega_k = \Omega_{k-1} \cup \{i_k\};$
     $\bar{\Omega}_k = \bar{\Omega}_{k-1} \setminus \{i_k\};$
     $z_k = Q_{k-1}^T a_{i_k};$
     $a_{i_k}^\perp = a_{i_k} - Q_{k-1} z_k;$
     $R_k = \begin{bmatrix} R_{k-1} & z_k \\ 0 & \|a_{i_k}^\perp\| \end{bmatrix};$
     $Q_k = \begin{bmatrix} Q_{k-1} & \frac{a_{i_k}^\perp}{\|a_{i_k}^\perp\|} \end{bmatrix};$
     $y_k = \begin{bmatrix} y_{k-1}, & \frac{\phi_{i_k}}{\|a_{i_k}^\perp\|} \end{bmatrix};$
     $r_k = r_{k-1} - \frac{\phi_{i_k}}{\|a_{i_k}^\perp\|^2} a_{i_k}^\perp;$
end for
return x with components $x(\Omega_K) = R_K^{-1} y_K$ and $x(\bar{\Omega}_K) = 0.$

---

parameter τ is formulated as

$$\min_x \|Ax - b\|^2 \quad \text{s.t.} \quad \|x\|_1 \leq \tau. \quad (2.62)$$

An alternative formulation is to minimize the l_1 -norm of the solution, which is known as the basis pursuit denoising problem

$$\min_x \|x\|_1 \quad \text{s.t.} \quad \|Ax - b\|^2 \leq \sigma. \quad (2.63)$$

Problems (2.62) and (2.63) are equivalent for an appropriate choice of parameters τ and σ [66]. Moreover, they can be reformulated as l_1 -regularized problem with parameter λ

$$\min_x \|Ax - b\|^2 + \lambda \|x\|_1. \quad (2.64)$$

Under some assumptions on A , solutions to these convex problems are sparse and solve the original cardinality-constrained problem (2.53). Note, that l_2 -regularized problem, known as Tykhonov regularization, generally has dense solutions. For survey of optimization methods in compressed sensing, we refer reader, e.g., to [12, 65].

7 Multiobjective optimization

To make a reader of Paper III familiar with multiobjective optimization, we provide here a short introduction to this area that contains problem formulation in the unconstrained case and the optimality conditions.

A *multiobjective* problem is formulated as follows:

$$\min_{x \in \mathbb{R}^n} \{f_i(x)\}_{i=1}^N \quad (2.65)$$

where $N > 1$ objective functions are to be minimized. The solution of problem (2.65) is a, so-called, *Pareto set*.

To define this set, consider two points $x', x'' \in \mathbb{R}^n$. We say, that point x' *dominates* point x'' if

$$f_i(x') \leq f_i(x''), \quad \forall i = 1, \dots, N,$$

and there exists $1 \leq k \leq N$ such that

$$f_k(x') < f_k(x'').$$

Point $x^* \in R^n$ is called *Pareto optimal* if there is no other point in R^n that dominates it. The Pareto set is composed of

$$\{f_1(x^*), \dots, f_N(x^*)\},$$

where all x^* are Pareto optimal.

For an overview of optimization methods for solving (2.65), we refer reader to [20, 45].

Optimal Design of Filter Networks

The second part of this thesis deals with applied optimization problems related to design of filter networks in medical imaging. In this area, algorithms for noise reduction, segmentation and image registration (see, e.g., [23, 35, 68]) require multidimensional finite impulse response (FIR) filters. The design of filters is determined by two important characteristics in image processing: high image quality and low processing time. These two are conflicting objectives, because any improvement in the quality requires a longer processing time. Traditional filters are computationally very demanding for 3D and 4D image processing, which limits their practical usage. Filter networks represent an alternative approach which is attractive due to the ability to significantly lower the computation time while maintaining its reasonably high quality.

This chapter contains background material related to design of filter networks, which might be necessary for specialists in optimization for reading Papers II and III. We mainly used [29, 62] as a source for this overview. We also discuss properties of the multilinear least-squares problem and present some new unpublished results.

1 Design of FIR filters

In the thesis, we study FIR filters of dimensionality d that have two representation spaces: the spatial space and the Fourier space, both of dimension d . The spatial space is discretized with unit sampling distance and spatial coordinates $\xi \in \mathbb{Z}^d$, that is, all integer points in $\mathbb{R}^n$. The frequencies $u \in \mathbb{R}^d$ in the Fourier space are normalized with respect to the sampling distance and measured in cycles per sample. Therefore, the Fourier space is 2π -periodical and we consider frequencies from the ball of radius π in l_∞ -norm

$$U = \{u \in \mathbb{R}^d : \|u\|_\infty \leq \pi\}.$$

A filter has an impulse response $f(\xi) : \mathbb{Z}^d \rightarrow \mathbb{C}$ in the spatial space and a frequency response $F(u) : U \rightarrow \mathbb{C}$ in the Fourier space, which are related to

each other by the Fourier transform

$$F(u) = (\mathcal{F}(f))(u) = \sum_{\xi \in \mathbb{Z}^d} f(\xi) e^{-iu^T \xi} \quad (3.1)$$

and by the inverse Fourier transform

$$f(\xi) = (\mathcal{F}^{-1}(F))(\xi) = \frac{1}{(2\pi)^d} \int_U F(u) e^{iu^T \xi} du. \quad (3.2)$$

For practical applications, it is easier to describe desired properties of filter in the Fourier space. For instance, a 2D quadrature filter [39], which is typical for local structure analysis, has frequency response in the polar coordinates (ρ, ϕ) of u

$$F(u) = \begin{cases} e^{-(\frac{2}{B \ln 2})^2 \ln^2(\frac{\rho}{\rho_0})} \cos^{2A}(\phi - \phi_k), & \text{if } |\phi - \phi_k| < \frac{\pi}{2}, \\ 0, & \text{otherwise,} \end{cases}$$

where ρ_0 , B , ϕ_k and A are certain parameters.

An arbitrary frequency response generally requires infinitely many spatial coordinates to define the corresponding impulse response, according to (3.2). But in practice, it is possible to use only a limited number of spatial coordinates $\{\xi_k\}_{k=1}^N$. Consider a filter defined by coefficients $c = (c_1 \dots, c_N)^T \in \mathbb{C}^N$, such that

$$f(\xi) = \begin{cases} c_k, & \text{if } \xi = \xi_k, \\ 0, & \text{otherwise,} \end{cases} \quad (3.3)$$

then its frequency response is defined by (3.1) as

$$\tilde{F}(u) = \sum_{k=1}^N c_k e^{-iu^T \xi_k}, \quad (3.4)$$

which is an approximation to the desired frequency response $F(u)$.

Since the spatial support of $f(\xi)$

$$\text{supp}(f) = \{\xi \in \mathbb{Z}^d : f(\xi) \neq 0\}$$

is bounded in $\mathbb{Z}^d$, there exists a closed ball $B_n(0) \subset \mathbb{R}^d$ of radius n in l_∞ -norm and centered in the origin

$$B_n(0) = \{y \in \mathbb{R}^d : \|y\|_\infty \leq n\}, \quad (3.5)$$

such that

$$\text{supp}(f) \subset B_n(0), \quad \text{supp}(f) \cap \partial B_n(0) \neq \emptyset,$$

where $\partial B_n(0) = \{y \in B_n(0) : \|y\|_\infty = n\}$ is the boundary of the ball.

Number n defines the *spatial size* of filter. The spatial support defines the *sparsity pattern* of filter in $B_n(0)$, i.e., the spatial locations of its non-zero coefficients. We say, that filter is *dense*, if

$$\text{supp}(f) = B_n(0) \cup \mathbb{Z}^d$$

and it has $N = (2n - 1)^d$ non-zero coefficients. Otherwise, it is a *sparse* filter of *sparsity level* $N < (2n - 1)^d$.

Given a sparsity pattern, we associate the filter design with the problem of finding best coefficients c , which is formulated as the weighted least-squares error problem

$$\min_c \int_U \left| F_w(u)(F(u) - \tilde{F}(u)) \right|^2 du. \quad (3.6)$$

Here $F_w(u) : U \rightarrow \mathbb{R}$ is a weighting function that enforces better approximation for selected frequencies. Note, that in general there exist other formulations of optimization problems related to filter design, which use ideal functions and weighting function defined in other spaces, e.g., in the spatial domain [40].

To treat numerically problem (3.6), the set U is sampled in m points $u_1, \dots, u_m$. In our papers, we used the regular Cartesian grid points. The number of sampling points along each coordinate axis in the Fourier space should be at least twice the number of sampling points in $\mathcal{N}$, i.e., $m > (4n)^d$. Then problem (3.6) is reduced to the linear least-squares problem

$$\min_{x \in \mathbb{C}^N} \|W(b - Ax)\|^2, \quad (3.7)$$

where $x = c$, $A \in \mathbb{C}^{m \times N}$, $b \in \mathbb{C}^m$ such that

$$A_{jk} = e^{-iu_j^T \xi_k}, \quad b_j = F(u_j) \quad (3.8)$$

for $j = 1, \dots, m$ and $k = 1, \dots, N$. The weighting matrix $W \in \mathbb{R}^{m \times m}$ is defined as

$$W_{ij} = \begin{cases} F_w(u_i), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

Here and further in the chapter $\|\cdot\|$ denotes the Euclidean norm.

For simplicity, we further assume $W = I$, but our reasoning remains valid for the general choice of W . We also assume that A , b and x are real-valued. Indeed, by splitting them into the real and imaginary parts we obtain

$$\|b - Ax\|^2 = \left\| \begin{pmatrix} \text{re}(b) \\ \text{im}(b) \end{pmatrix} - \begin{bmatrix} \text{re}(A) & -\text{im}(A) \\ \text{im}(A) & \text{re}(A) \end{bmatrix} \begin{pmatrix} \text{re}(x) \\ \text{im}(x) \end{pmatrix} \right\|^2.$$

Other design preferences, like symmetry of filter or exact zero response in the origin, can be implemented with the use of an appropriate transformation of

variables [50]. Hence, by properly changing notations, we associate design of FIR filters with the real-valued linear least-squares problem

$$\min_{x \in \mathbb{R}^N} \|b - Ax\|^2, \quad (3.9)$$

where $A \in \mathbb{R}^{m \times N}$, $b \in \mathbb{R}^m$ and the number of the decision variables is closely related to the number of non-zero coefficients of the filter.

On the one side, it is clear, that the larger number of coefficients we use, the smaller would be the approximation error in (3.9), and hence, better image processing quality. For illustration, consider in Figure 1 a 2D band-pass quadrature filter sampled in $m = 361$ points and its approximation by a dense filter and a sparse filter of spatial size 9 with 81 and 41 non-zero coefficients, respectively.

Figure 1: Frequency response of a) ideal 2D band-pass quadrature filter b) dense filter and c) sparse filter

On the other side, this increases the processing time, because the time is proportional to the number of arithmetical operations in convolution of image with filter, which, in turn, is proportional to the number of non-zero coefficients. Indeed, let $g : \mathbb{Z}^d \rightarrow \mathbb{R}$ denote the image response in the spatial domain, then the result of its convolution with filter $f(\xi)$ is computed by formula

$$(f * g)(\xi) = \sum_{\zeta \in \mathbb{Z}^d} f(\xi + \zeta)g(\xi - \zeta) = \sum_{k=1}^N c_k g(\xi - \zeta_k), \quad (3.10)$$

where $\zeta_k = \xi_k - \xi$, which follows from the filter definition in the spatial space (3.3).

The traditional approach in image processing is to use a dense filter, which ensures the best approximation. But convolution with image (3.10) is computationally expensive and complexity grows exponentially with the signal dimension. By using one sparse filter instead of the dense one, we can improve the processing time, but this increases the approximation error and degrades the image quality.

Filter networks represent an approach that consists in a decomposition of a dense filter into several filters of smaller spatial size connected in a certain way. We refer to them as sub-filters. In the next section, we describe how filter networks perform and what decisions should be made in the process of their design.

2 Filter networks

The idea of filter networks is based on the following observation. Consider two sequentially connected sub-filters $f_1(\xi)$ and $f_2(\xi)$ of dimensionality d and spatial size n_1 and n_2 with N_1 and N_2 non-zero coefficients, respectively, where $N_i \ll n_i^d$ for $i = 1, 2$. Their convolution generates a filter of spatial size $n_1 + n_2 - 1$ with frequency response

$$\mathcal{F}(f_1 * f_2) = \mathcal{F}(f_1) \cdot \mathcal{F}(f_2) \quad (3.11)$$

according to the convolution theorem [4].

The new filter is typically dense and can give a good approximation to the desired frequency response. But instead of convolving image with the dense filter, it is convolved sequentially with the sub-filters, which allows to save approximately $(2n_1 + 2n_2 - 3)^d / (N_1 + N_2)$ operations.

Filter networks can be represented as a directed acyclic graph, whose nodes stand for sub-filters [63]. As we have shown above, sequentially connected sub-filters act as a single dense filter, which allows to approximate the desired frequency response to a high accuracy. But the total number of non-zero coefficients in the filter network is much less than in the corresponding single dense filter. Therefore, filter networks are computationally much more efficient, which makes them attractive for using in practice.

Design of filter networks is related to non-trivial optimization problems, which limits their usage in practice. Experts in filter design are required to choose a network structure, i.e., a number of sub-filters and their connection, a sparsity pattern for each sub-filter and values of their coefficients. These choices correspond to solving some optimization problems of multi-extremal and multi-criterial nature, as we will further demonstrate. For this reason, the decisions on a network structure and its sparsity pattern were mainly based on the individual expertise of specialists in filtering and their intuition.

An overview of early attempts to design filter networks is given in [58]. Another approach considers low-rank approximations of a multidimensional filter by a sequence of 1D filters. It is based on the singular value decomposition techniques [27]. For an arbitrary filter, the quality of low-rank approximation can be very low, unless it is nearly Cartesian separable. For instance, Gaussian functions are Cartesian separable and can be used in filter design. However, they are typically less efficient for some applications in medical imaging, like analyzing local structure.

In view of this, we were motivated to develop optimization methods for design of filter networks of a general structure and frequency response. In the next section, we study multilinear least-squares (MLLS) problem related to design of sequentially connected sub-filters with fixed sparsity pattern, which represent the simplest class of filter networks.

3 Multilinear least-squares problem

In the case of the sequential connection of L filters, let matrices $A_i \in \mathbb{R}^{m \times n_i}$, $i = 1, 2, \dots, L$ be determined by the Fourier transform for sub-filter i in the way described in section 1 and [50]. Note, that the sparsity pattern is implicitly defined in A_i , whose columns correspond to the locations of non-zero coefficients.

For a given desired frequency response $b \in \mathbb{R}^m$, the coefficients for the filter network can be found by solving the MLLS problem

$$\min_{x \in \mathbb{R}^N} \|b - (A_1 x_1) \circ (A_2 x_2) \circ \dots \circ (A_L x_L)\|^2 \equiv \|r(x)\|^2, \quad (3.12)$$

where $x_i \in \mathbb{R}^{n_i}$, $i = 1, 2, \dots, L$, $N = n_1 + n_2 + \dots + n_L$ and $x = (x_1^T, x_2^T, \dots, x_L^T)^T$. Here $u \circ v$ denotes the component-wise product of vectors u and v . The vector

$$(A_1 x_1) \circ (A_2 x_2) \circ \dots \circ (A_L x_L) \equiv A(x) \in \mathbb{R}^m$$

represents the frequency response of the filter network, which follows from (3.11).

For $L > 1$, MLLS is a non-convex optimization problem. The numerical experiments show that the number of local minima of the objective function maybe very large. Moreover, each local minimizer is singular and non-isolated. Indeed, if $x_1, \dots, x_L$ is a local minimizer, then $t_1 x_1, \dots, t_L x_L$ is also a local minimizer for any combination of scalars $t_1, \dots, t_L$ such that $t_1 \cdots t_L = 1$. This means, that each local minimizer belongs to a surface of local minimizers. Typical values of its parameters are $m \in [10^d, 40^d]$, $L \in [2, 10]$, $n_i \in [3^d, 15^d]$. This makes MLLS to be a large-scale problem in case of design of 3D and 4D filters. Moreover, even in 2D case, the number of internal parameters m is very large.

Analysis of this problem shows, that the Jacobian $J(x) \in \mathbb{R}^{m \times N}$ of $r(x)$ is rank-deficient, which follows from the next result.

Lemma 2 *Let $x \in \mathbb{R}^N$ be such that $A(x)$ has no zero components. Then*

$$\text{rank}(J(x)) \leq N - L + 1. \quad (3.13)$$

Proof. The Jacobian of $r(x)$ can be represented as

$$J(x) = \text{diag}(A(x)) \left[\frac{A_1}{A_1 x_1} \cdots \frac{A_L}{A_L x_L} \right], \quad (3.14)$$

where for convenience we use notation

$$\frac{A_i}{A_i x_i} \equiv \text{diag}(A_i x_i)^{-1} A_i.$$

Consider the set of vectors

$$u_i = (x_1^T, 0, \dots, 0, -x_i^T, 0, \dots, 0)^T \in \mathbb{R}^N, \quad i = 2, \dots, L, \quad (3.15)$$

then

$$J(x)u_i = \text{diag}(A(x)) \left[\frac{A_1 x_1}{A_1 x_1} - \frac{A_i x_i}{A_i x_i} \right] = 0, \quad \forall i \geq 2.$$

Clearly, for any linear combination of these vectors $u \in \text{span}(u_2, \dots, u_L) = S_{L-1}$ we have

$$J(x)u = 0.$$

Therefore, to obtain the upper bound (3.13) on the rank of $J(x)$, it is sufficient to show that

$$\dim(S_{L-1}) = L - 1. \quad (3.16)$$

Suppose, to the contrary, that $u_2, \dots, u_L$ are linearly dependent and there exist $(\lambda_1, \dots, \lambda_{L-1}) \in \mathbb{R}^{L-1}$ such that

$$\sum_{i=1}^{L-1} \lambda_i u_{i+1} = 0, \quad (3.17)$$

$$\sum_{i=1}^{L-1} \lambda_i \neq 0. \quad (3.18)$$

Equality (3.17) combined with (3.15) implies

$$x_1 \left(\sum_{i=1}^{L-1} \lambda_i \right) = 0.$$

Due to (3.18), this would give $x_1 = 0$, which implies $A_1 x_1 = 0$. This contradicts the assumption that $A(x)$ has no zero components, which proves (3.16) and accomplishes the proof of lemma.

□

It remains an open question whether bound (3.13) is tight, and if not, then what is the lower bound.

MLLS problem belongs to the class of nonlinear least-squares problems, which was presented in section 5 of Chapter 2. Conventional local optimization methods require generating randomly a number of starting points for their further refinement to solve MLLS. We observed in practice two major drawbacks of this

approach. Firstly, a very large number of starting points is typically required to be generated in order to find a reasonably good solution to problem (3.12). Secondly, the convergence of local methods is too slow in this problem.

In practice, the most popular of the local algorithms for solving the MLLS problem is the alternating least squares (ALS) method. In literature, it is also known as block-coordinate relaxation or nonlinear Gauss-Seidel algorithm [51]. ALS exploits the special structure of the objective function in the MLLS problem and iteratively improves solution x by solving a series of linear least-squares problems resulting in minimizing $f(x)$ over the segment x_i for alternating index i while the rest of the segments of x remain fixed. It is formally presented in Algorithm 8. This algorithm is simple to implement, however, the mentioned above major drawbacks of the local search algorithms are present in ALS as well.

Algorithm 8 Alternating Least-Squares

Require: $x \in \mathbb{R}^N$.
repeat
    for $i = 1, \dots, L - 1, L, L - 1, \dots, 2$ **do**
         $D_i = \prod_{j \neq i} \text{diag}(A_j x_j)$;
         $x_i = (D_i A_i)^\dagger b$;
    end for
until stopping criteria met

The success of local optimization methods, including ALS, depends on the choice of the starting point. In the next subsections, we discuss two approaches for selecting a good starting point.

3.1 Global search approach

The first approach was present in [50], where the following problem was introduced

$$\begin{aligned} \min_{x \in \mathbb{R}^N, y \in \mathbb{R}^{mL}} \quad & \sum_{i=1}^L \|y_i - A_i x_i\|^2, \\ \text{s.t.} \quad & y_1 \circ \dots \circ y_L = b \end{aligned}$$

where $y = (y_1^T, y_2^T, \dots, y_L^T)^T$. The new problem is conceptually close to MLLS and its solution generates a good starting point for ALS. It can be equivalently reformulated as

$$\begin{aligned} \min_{y \in \mathbb{R}^{mL}} \quad & \sum_{i=1}^L y_i^T P_i y_i \\ \text{s.t.} \quad & y_1 \circ \dots \circ y_L = b, \end{aligned} \tag{3.19}$$

where $P_i = I - A_i A_i^\dagger$ is a projection matrix defined by A_i . In Paper II, we proposed a special procedure for moving from one local solution to a better one.

We proved the correctness of this approach theoretically, and we observed its efficiency on test problems originating from design of filter networks.

Figure 2 presents results of solving MLLS problem related to design of 2D band-pass quadrature filter. We used different starting points for ALS: 50 randomly generated starting points (grey lines), one deterministic point as defined in [50] (dashed line) and a point generated by solving (3.19) (solid black). The approximation error ε^2 , which is the value of the objective function in (3.12), is given in the logarithmic scale on base 10 as function of the number of iterations, where one ALS iteration corresponds to the run of a single for-loop in Algorithm 8. In case of a randomly generated starting point, it was typical for ALS to stagnate in a surface of local minimizers for a number of iterations without any improvement in the objective function which followed by a "jump" to a new local minimizer with a better objective function value. The starting point generated by our approach was refined by ALS without jumps and the obtained solution was among the most successful ones.

Figure 2: Performance of ALS executed from various starting points to solve MLLS problem related to design of 2D quadrature band-pass filter of order 2. Grey lines: 50 randomly generated starting points. Dashed line: the deterministic starting point. Solid black line: the starting point generated by the global search approach

3.2 Continuation approach

An alternative approach to generate a good starting point for solving MLLS is based on the idea of continuation methods [2] widely used, e.g., for solving optimization problems and systems of nonlinear equations, especially when the solutions are not unique. Continuation methods define an easy problem for which we know the solution, and a path between this easy problem and the hard problem that we actually wish to solve. The solution to the easy problem is gradually transformed to the solution of the hard problem by tracing this path.

In the case of problem (3.12), the path may be defined by introducing an additional scalar parameter t into the problem as follow

$$\min_x \|tb + (1-t)b_0 - (A_1x_1) \circ (A_2x_2) \circ \cdots \circ (A_Lx_L)\|^2, \quad (3.20)$$

where $b_0 = (A_1x_1^0) \circ \cdots \circ (A_Lx_L^0)$ is calculated for a given initial point $x_1^0, \dots, x_L^0$.

Note that if $t = 0$ it is an easy problem. For $t = 1$, it is equivalent to (3.12). We start from $t = 0$ and solve (3.20) for a slightly increased value of t . Since it is a small perturbation to the problem, the local search methods, like ALS, can be successfully used with the initial point x^0 . Then t is slightly increased again and the chosen local search method is applied starting from the solution obtained for the previous value of t . Alternatively, a predicted step can be computed first as described in [2] and then refined by ALS. In this way we can reach a local solution to (3.20).

In our numerical experiments, the continuation approach for some cases could significantly improve solution obtained by running ALS from $x(1)$ when compared with the solution obtained from $x(0)$. Unfortunately, this approach was not related with a substantial improvement.

This is a very simple version of the continuation methods. The most advanced of them allows to move from one local solution of (3.12) to another by following a path parametrized by a new variable $s \in \mathbb{R}$. In case of MLLS problem, consider

$$H(x, t) = tb_0 + (1-t)b - (A_1x_1) \circ \cdots \circ (A_Lx_L) = 0,$$

where

$$\begin{cases} x = x(s), \\ t = t(s). \end{cases}$$

By taking the total derivative of $H(x(s), t(s))$ with respect to s , we obtain

$$\frac{\partial H}{\partial x} \frac{dx}{ds} + \frac{\partial H}{\partial t} \frac{dt}{ds} = 0. \quad (3.21)$$

Denote

$$(\dot{x}, \dot{t}) = \left(\frac{dx}{ds}, \frac{dt}{ds} \right),$$

which is a tangent vector to the path in $\mathbb{R}^{N+1}$. Then equation (3.21) can be viewed as an underdetermined system of linear equations with the matrix of size $N \times (N + 1)$. For this reason, it is augmented with an extra equation, e.g., the one that normalizes the length of this vector

$$\|\dot{x}\|^2 + |\dot{t}|^2 = 1. \quad (3.22)$$

This normalization permits interpreting s as the true arc along the path from $(x^0, 0)$ to $(x(s), t(s))$.

By solving the new linear system of full rank composed of (3.21) and (3.22), one can go along the continuation path and revisit several times $x(1)$, which is a local solution to MLLS. In future, we plan to implement this approach.

Summary of Papers

The purpose of this chapter is to give an overview of the results in the contributed to the thesis papers and discuss future work.

In **Paper I**, our aim was to develop efficient methods for solving a general large-scale unconstrained problem. Limited memory quasi-Newton methods and trust region techniques are well-known to be effective in unconstrained optimization. But their straightforward combination deteriorates the efficiency of the former approach. Therefore, in practice limited memory quasi-Newton methods are typically combined with a line-search.

We considered limited memory quasi-Newton updates that admit a compact representation (2.43). Such representation holds, e.g., for BFGS (2.44), SR 1 (2.45) and multipoint symmetric secant approximations (2.46). Some of our approaches were based on the eigenvalues of the Hessian approximation. We derived the following representation of the eigenvalue decomposition:

$$B = P \begin{pmatrix} \Lambda & 0 \\ 0 & \delta I_{n-\bar{m}} \end{pmatrix} P^T,$$

where index k is omitted for simplicity, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{\bar{m}})$ and $P = [P_{\parallel} \ P_{\perp}] \in R^{n \times n}$ is an orthogonal matrix with $P_{\parallel} \in R^{n \times \bar{m}}$ and $P_{\perp} \in R^{n \times (n-\bar{m})}$. The main assumption here is that $\bar{m} \ll n$.

There are at most $\bar{m} + 1$ distinct eigenvalues computed at a cost of $O(\bar{m}^3)$ operations, while the eigenvectors are defined implicitly. This allows to find a nearly-exact solution to the trust-region subproblem defined in the Euclidean norm at a cost of $2\bar{m}n + O(\bar{m})$ operations.

The other approach is based on the eigenvalue-based norms

$$\begin{aligned} \|s\|_{P,\infty} &\equiv \max \left(\|P_{\parallel}^T s\|_{\infty}, \|P_{\perp}^T s\|_2 \right), \\ \|s\|_{P,2} &\equiv \max \left(\|P_{\parallel}^T s\|_2, \|P_{\perp}^T s\|_2 \right). \end{aligned}$$

In the new norms, the trust-region subproblem is separable and each of the smaller subproblems is easy to solve.

We implemented the proposed methods using the limited memory BFGS update. The computational cost of one iteration of our algorithms is at most $2\bar{m}n +$

$O(\bar{m}^3)$ with the highest term the same as for computing the search direction in the line-search L-BFGS algorithms. Moreover, we used the proposed updating techniques to improve existing versions of limited memory trust-region methods.

The reported results of the numerical experiments demonstrate that our limited-memory trust-region algorithms are competitive with the line-search version of the L-BFGS method.

In future, we plan to extend these approaches to limited memory SR1 and multipoint symmetric secant approximations. They are better suited for the trust-region framework, because in case of indefinite matrix in the trust-region subproblem, one can exploit the useful information about negative curvature directions. The other possibility is to extend our techniques to limited-memory algorithms for solving constrained and bound constrained optimization problems.

In **Paper II**, our aim was to develop optimization tools that allow for efficiently solving the MLLS problem. This problem is related to the design of sequential filter networks with fixed sparsity pattern. The success of local optimization methods, like ALS, depends on the choice of the starting point. To generate such point, we studied a new constrained problem (3.19) which is similar, in some sense, to MLLS. The new problem consists of a finite number of easy-to-solve subproblems, which can be enumerated by sign combinations. In Theorem 1, we derived optimality conditions to check if the current combination of signs is optimal. Based on this result, we proposed a global search strategy which successfully improves the sign combination and decreases the objective function in (3.19).

The numerical experiments in design of 2D filters demonstrated efficiency of the proposed algorithm, which is not computationally very demanding. The solution to the MLLS problem produced by ALS from starting point generated by our global approach, was as good as the best of those produced by ALS from 500 random starting points.

In **Paper III**, our aim was to develop optimization tools for finding sparsity pattern for sub-filters that can be used in design of filter networks.

For this reason, we studied the case of sequential filter networks and formulated cardinality-constrained MLLS problem

$$\min_{\|x\|_0 \leq K} \|b - (A_1 x_1) \circ (A_2 x_2) \circ \dots \circ (A_L x_L)\|^2,$$

where the cardinality constraint bounds above the number of non-zero coefficients in the network.

We proposed an approach for approximately solving this problem. It exploits the special structure of the objective function that allows for reducing the problem to a sequence of relatively easier cardinality-constrained linear least-squares problems. It was then applied to solving a bi-criteria optimization problem in

---

which both the time and quality of image processing are optimized. The resulting Pareto set approximation offers the practitioners a means of finding a proper trade-off between the image processing quality and time.

The developed approach was extended to designing filter networks of a more general structure. Its efficiency was demonstrated by designing some 2D and 3D filter networks. We also compared it with the existing approaches. We observed that it allows also for improving network structure, which makes automated some stages of designing filter networks.

The presented results demonstrated some of the high potentials of our approach. In the future, more refined implementations will be able to demonstrate in full scale its efficiency.

Finally, we note that the range of possible applications goes beyond the design of filter networks. Similar problems occur, e.g., in factor analysis, chemometrics, psychometrics ([42], [43], [44], [52], [67]). Our approach may be extended to solving such problems.

References

- [1] M. Al-Baali, E. Spedicato, and F. Maggioni. Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: a review and open problems. *Optimization Methods and Software*, (ahead-of-print):1–18, 2013.
- [2] E. Allgower, L. Eugene, and K. Georg. *Introduction to numerical continuation methods*, volume 45. SIAM Classics in Applied Mathematics, 2003.
- [3] T. Blumensath and M.E. Davies. Iterative hard thresholding for compressed sensing. *Appl Comput Harmon Anal*, 27(3):265 – 274, 2009.
- [4] R.N. Bracewell and R.N. Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.
- [5] O.P. Burdakov. Methods of the secant type for systems of equations with symmetric Jacobian matrix. *Numerical functional analysis and optimization*, 6(2):183–195, 1983.
- [6] O.P. Burdakov. Stable versions of the secants method for solving systems of equations. *USSR Computational Mathematics and Mathematical Physics*, 23(5):1–10, 1983.
- [7] O.P. Burdakov. On superlinear convergence of some stable variants of the secant method. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 66(12):615–622, 1986.
- [8] O.P. Burdakov, J.M. Martinez, and E.A. Pilotta. A limited-memory multi-point symmetric secant method for bound constrained optimization. *Annals of Operations Research*, 117:51–70, 2002.
- [9] J.V. Burke, A. Wiegmann, and L. Xu. Limited memory BFGS updating in a trust-region framework. Technical report, University of Washington, 2008.
- [10] R.H. Byrd, J. Nocedal, and R.B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156, 1994.
- [11] R.H. Byrd, R.B. Schnabel, and G.A. Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical programming*, 40(1-3):247–263, 1988.
- [12] E.J. Candès and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

-
- [13] S. Cheng and N. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. *SIAM Journal on Matrix Analysis and Applications*, 19(4):1097–1110, 1998.
  - [14] A.R. Conn, N.I.M. Gould, and Ph.L. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50(1-3):177–195, 1991.
  - [15] A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *Trust-Region Methods*. MPS/SIAM Ser. Optim. MP01, SIAM, Philadelphia, 2000.
  - [16] Y. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on Optimization*, 10(1):177–182, 1999.
  - [17] R. Dembo, S. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.
  - [18] J.E. Dennis and H.H.W. Mei. Two new unconstrained optimization algorithms which use function and gradient values. *Journal of Optimization Theory and Applications*, 28(4):453–482, 1979.
  - [19] J.E. Dennis and R.B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16. SIAM, 1983.
  - [20] M. Ehrgott. *Multicriteria optimization*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
  - [21] J.B. Erway and R.F. Marcia. Limited-memory BFGS systems with diagonal updates. *Linear Algebra and its Applications*, 437(1):333–344, 2012.
  - [22] J.B. Erway and R.F. Marcia. MSS: MATLAB software for L-BFGS trust-region subproblems for large-scale optimization. *To appear in ACM Transactions on Mathematical Software*, 2013.
  - [23] R.S.J. Estepar. *Local structure tensor for multidimensional signal processing. Applications to medical image analysis*. Presses universitaires de Louvain, Louvain-la-Neuve,Belgium, 2007.
  - [24] A.V. Fiacco and G.P. McCormick. Wiley, New York, 1968. Reprinted by SIAM Publications, 1990.
  - [25] R. Fletcher and C.M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.
  - [26] P.E. Gill, W. Murray, and M.H. Wright. *Practical optimization*. London: Academic Press, 1981.
  - [27] G.H. Golub and Ch.F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 3rd edition, 1996.

- [28] N.I.M. Gould, D. Orban, and Ph.L. Toint. Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, 14:299–361, 4 2005.
- [29] G. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer, Dordrecht, 1995.
- [30] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM e-books. SIAM, Philadelphia, 2nd edition, 2008.
- [31] W. Hager. Minimizing a quadratic over a sphere. *SIAM Journal on Optimization*, 12(1):188–208, 2001.
- [32] W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.
- [33] W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.
- [34] W. Hager and H. Zhang. The limited memory conjugate gradient method. Technical report, 2012.
- [35] M. Hemmendorff, M.T. Andersson, T. Kronander, and H. Knutsson. Phase-based multidimensional volume registration. *Medical Imaging, IEEE Transactions on*, 21(12):1536–1543, 2002.
- [36] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems1. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.
- [37] L. Kaufman. Reduced storage, quasi-Newton trust region approaches to function optimization. *SIAM Journal on Optimization*, 10(1):56–69, 1999.
- [38] H. Khalfan, R.H. Byrd, and R.B. Schnabel. A theoretical and experimental study of the symmetric rank-one update. *SIAM Journal on Optimization*, 3(1):1–24, 1993.
- [39] H. Knutsson. Representing Local Structure Using Tensors. In *Proceedings of the 6th Scandinavian Conference on Image Analysis*, volume 1019 of *LiTH-ISY-I*, pages 244–251, Linköping, 1989. Linköping University Electronic Press.
- [40] H. Knutsson, M. Andersson, and J. Wiklund. Advanced filter design. In *Proceedings of the 11th Scandinavian Conference on Image Analysis : Greenland*, pages 185–193. SCIA, 1999.
- [41] T.G. Kolda. *Limited-Memory Matrix Methods with Applications*. PhD thesis, University of Maryland, 1997.

-
- [42] R. Leardi, C. Armanino, S. Lanteri, and L. Alberotanza. Three-mode principal component analysis of monitoring data from venice lagoon. *Journal of Chemometrics*, 14(3):187–195, 2000.
- [43] S. Leurgans and R.T. Ross. Multilinear models: applications in spectroscopy. *Statistical Science*, 7(3):289–310, 1992.
- [44] J.A. Lopes and J.C. Menezes. Industrial fermentation end-product modelling with multilinear PLS. *Chemometrics and Intelligent Laboratory Systems*, 68(1):75–81, 2003.
- [45] K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, USA, 1999.
- [46] J.J. Moré and D. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.
- [47] S. Muthukrishnan. *Data streams: algorithms and applications*. Now Publishers, Boston, 2005.
- [48] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.
- [49] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer-Verlag, New York, USA, 2nd edition, 2006.
- [50] B. Norell, O. Burdakov, M. Andersson, and H. Knutsson. Approximate spectral factorization for design of efficient sub-filter sequences. Technical Report LiTH-MAT-R-2011-14, Department of Mathematics, Linköping University, 2011.
- [51] J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, reprint of the 1970 original edition, 2000.
- [52] P. Paatero. Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37(1):23 – 35, 1997.
- [53] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969.
- [54] M.J.D. Powell. A hybrid method for nonlinear equations. *Numerical methods for nonlinear algebraic equations*, 7:87–114, 1970.
- [55] M.J.D. Powell. *A new algorithm for unconstrained optimization*. UKAEA, 1970.
- [56] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming*, 77(1):273–299, 1997.

- [57] M. Rojas, S. Santos, and D. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM Journal on Optimization*, 11(3):611–646, 2001.
- [58] T. Sarmaki and A.T. Fam. Subfilter approach for designing efficient FIR filters. In *Circuits and Systems, 1988., IEEE International Symposium on*, pages 2903–2915 vol.3, 1988.
- [59] D. Sorensen. Minimization of a large-scale quadratic functions subject to a spherical constraint. *SIAM Journal on Optimization*, 7(1):141–161, 1997.
- [60] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.
- [61] W. Sun and Y. Yuan. *Optimization theory and methods: nonlinear programming*, volume 98. Springer, 2006.
- [62] B. Svensson. *A Multidimensional Filtering Framework with Applications to Local Structure Analysis and Image Enhancement*. PhD thesis, Linköping University, Medical Informatics, The Institute of Technology, 2008.
- [63] B. Svensson, M. Andersson, and H. Knutsson. A graph representation of filter networks. In Heikki Kalviainen, Jussi Parkkinen, and Arto Kaarna, editors, *Image Analysis*, volume 3540 of *Lecture Notes in Computer Science*, pages 1086–1095. Springer Berlin Heidelberg, 2005.
- [64] Ph.L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. *Sparse matrices and their uses*, pages 57–88, 1981.
- [65] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. In *Proceedings of the IEEE*, volume 98(6), pages 948–958, 2010.
- [66] E. van den Berg and M. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2009.
- [67] J.H. Wang, P.K. Hopke, T.M. Hanczewicz, and S.L. Zhang. Application of modified alternating least squares regression to spectroscopic image analysis. *Analytica Chimica Acta*, 476(1):93–109, 2003.
- [68] C.-F. Westin, L. Wigström, T. Looock, L. Sjöqvist, R. Kikinis, and H. Knutsson. Three-dimensional adaptive filtering in magnetic resonance angiography. *Journal of Magnetic Resonance Imaging*, 14(1):63–71, 2001.
- [69] Ph. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

Papers

The articles associated with this thesis have been removed for copyright reasons. For more details about these see:

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-103646>