Technische Universität München

Department of Mathematics

Bundle Methods for Nonsmooth, Nonconvex Problems with Inexact Information

Master's Thesis

by

Annika Stegie

Supervisor:	Prof. Dr. Michael Ulbrich
Advisers:	Dr. Andre Milzarek Lukas Hertlein (M.Sc.)
Submission date:	Datum

I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Garching, Date

Acknowledgments

First of all I thank Prof. Ulbrich for the opportunity to write my thesis at his chair. I thank my advisers for the the interesting topic and the good support and guidance in the course of this thesis. Particularly I want to thank Dr. Andre Milzarek for the good communication even over longer distances and Lukas Hertlein for taking over the mentoring so naturally.

Abstract

Bundle methods are very popular methods when it comes to the solution of nonsmooth optimization problems. To apply them to various kinds of real world problems it is important that they are able to work with nonconvex objective functions as well as inexact evaluations of the function value and subgradient. A survey on the different strategies that are used to tackle these issues is presented in this thesis.

A main contribution of this work is the development of a variable metric variant of the bundle method that can make use of curvature information. It is compared to another bundle method suitable for noisy nonconvex objective functions. Finally the performance of bundle methods for the solution of bilevel problems is explored. This is done by using the method for the hyper-parameter optimization for a support vector classifier.

German Summary

In dieser Arbeit wird die Klasse der Bundle-Methoden im Hinblick auf nichtkonvexe Zielfunktionen, inexakte Funktionen- und Subgradientenauswertungen und die Möglichkeit der Nutzung von Krümmungsinformation untersucht. Bundle Methoden werden sehr erfolgreich zur Optimierung von nichtdifferenzierbaren Funktionen eingesetzt, vor allem die Anwendung unter Inexaktheit ist jedoch eine neuere Entwicklung.

Es werden zunächst verschiedene Strategien vorgestellt, die es möglich machen, Bundle Algorithmen zur Lösung nichtkonvexer und nicht exakter Funktionen zu verwenden. Ein wichtiger Teil der Arbeit besteht außerdem in der Entwicklung einer Variante der Bundle Methode, die auch eventuell vorhandene Krümmungsinformationen der Zielfunktion nutzt. Ein Vergleich der Methode mit einem proximalen Bundle Algorithmus zeigt das Potential dieser Variante auf, welches jedoch je nach Anwendung stark variiert.

Schließlich wird noch die Anwendbarkeit von Bundle Methoden auf Bilevel Probleme betrachtet. Dazu wird ein Bundle Algorithmus auf das Problem der Parameteroptimierung für einen Support Vector Klassifizierer angewendet.

Contents

List of Symbols

α_j^k	≥ 0	Lagrange multipliers of the bundle subproblem
a_k		aggregate linearization
A_k		augmented aggregate linearization
b		bias (chapter 7)
b_j^k		convexification term of the augmented linearization error
$B_r(x)$		open ball with radius r around the point x
c_j^k	≥ 0	augmented linearization error
C		hyper-parameter (chapter 7)
C^k	≥ 0	augmented aggregate error
d^k		step in bundle method / minimizer of the bundle subproblem
δ_k		decrease measure
δ_k^M	≥ 0	model decrease
e_j^k	≥ 0	linearization error
$\tilde{e}_j^k$	≥ 0	subgradient locality measure
E^k	≥ 0	aggregate (linearization) error
η_k	≥ 0	convexification parameter
$f(x)$		exact evaluation of f at point x
f_k		value of f at point x^k from the oracle, may be inexact
$\hat{f}_k$		value of f at stability center $\hat{x}^k$, may be inexact
g^k		subgradient at point x^k, can be exact or inexact
G		number of groups (chapter 7)
G^k		aggregate subgradient
γ	> 0	safeguarding parameter for η-calculation; in chapter 7: margin
$\mathtt{i}_X$		indicator function of the set X
$\mathbb{I}$		identity matrix
J_k		bundle index set at iteration k
$\mathcal{J}F$		Jacobian of function F (chapter 7)
$\mathcal{J}_w F$		partial Jacobian of F with respect to w (chapter 7)
$\kappa_{+,-}$		step size updating parameters
l_j		linear function (cutting plane)
L		Lagrangian (chapter 7)
$\mathcal{L}$		loss function (chapter 7)
λ		hyper-parameter in risk minimization (chapter 7)
m	$\in (0,1)$	decrease parameter

m_k		cutting plane model
$\tilde{m}_k$		regularized cutting plane model
M_k		(augmented) cutting plane model of the convexified objective function
μ		Lagrange multiplier (chapter 7)
n_d		number of data points (chapter 7)
n_f		size of feature space (chapter 7)
N_U		normal cone of set U (chapter 7)
$\bar{\mathcal{N}}$		index set of training data (chapter 7)
$\mathcal{N}$		index set of validation data
ν^k		subgradient of the constraint set
Q_k		metric matrix
s^k		augmented subgradient at x^k
S		solution map (chapter 7)
S^k		augmented aggregate subgradient
$\bar{\sigma}$		error bound on approximate function value
t_k	> 0	prox-parameter
T		number of folds (chapter 7)
$\bar{\theta}$		error bound on approximate subgradient
U_{ad}		feasible set (chapter 7)
$w, \tilde{w}$		separating hyperplane (chapter 7)
x^i		data point (chapter 7)
x^k		iterate of bundle algorithm
$\hat{x}^k$		current stability center
$\xi, \tilde{\xi}$		slack variable from reformulation of subproblems
y_i		class label (chapter 7)
∇f		gradient of function f (chapter 7)
$\nabla_w f$		partial gradient of f with respect to w (chapter 7)
∂f		subdifferential of function f
$\partial_\varepsilon f$		ε-subdifferential of function f
$\partial_{[\varepsilon]} f$		Fréchet ε-subdifferential

List of Figures

List of Tables

1. Introduction

There exists a sound and broad theory of classical nonlinear optimization. However, this theory puts strong differentiability requirements on the given problem. Requirements that cannot always be fulfilled in practice. Examples for such nondifferentiable applications reach from problems in physics and mechanical engineering [4] over optimal control problems up to data analysis [2] and machine learning [53]. Other possible fields of applications are risk management and financial calculations [42, 57]. Additionally the problem class of bilevel programs can yield nonsmooth objective functions as shown in [48] and [41]. There is hence a need for nonsmooth optimization algorithms.

A lot of the underlying theory was developed in the 1970's also driven by the "First World Conference on Nonsmooth Optimization" taking place in 1977 [39]. These days, there exists a well understood theoretical framework of nonsmooth analysis to create the basis for practical algorithms [51].

The most popular methods to tackle nonsmooth problems at the moment are bundle methods [16]. First developed only for convex functions [29] these methods were soon extended to cope also with nonconvex objective functions [38]. Some time later the algorithms were again enhanced to deal with inexact information of the function value, the subgradient or both. Some natural applications for these cases are derivative free optimization and stochastic simulations [16].

The basic idea of bundle methods is to model the original problem by a simpler function, often some sort of stabilized cutting plane model, that is minimized as a subproblem of the algorithm [19, chapter XV]. The computed iterate is tested for sufficient descent and depending on the result is either taken as the new iterate or the model is enhanced.

There exist different types of bundle methods, a widely used one being the proximal bundle method. In this thesis two types of bundle methods are worked with. One is of the proximal type and one uses a variable stabilization term that makes it possible to make use of curvature information in order to accelerate the convergence speed. The development of the algorithm NOCHETWAS SCHREIBEN

The first half of this work puts particular attention on the theoretical concepts to use bundle methods with nonconvex and inexact objectives and how to incorporate the curvature information into the method.

In the second half of the thesis the usability of bundle algorithms for bilevel programs is explored. Bilevel problems consist of an upper level problem constrained by an additional

optimization problem, the lower level. These problems occur in a variety of applications such as game theory (see [5, section 2.1] for a variety of applications). Here the bilevel problem is derived from the hyper-parameter optimization for support vector machines. In the application both nonconvexity of the objective function and inexactness in the function value and subgradient calculation are addressed.

The remainder of the thesis is organized as follows: After a short introduction of the most important definitions and results from nonsmooth analysis in chapter 2 a basic bundle algorithm for exact convex functions is stated in order to introduce the important concepts of this method in chapter 3. A survey of different methods to tackle inexactness and nonconvex objective functions is then presented in chapter 4. Chapter 5 reviews the proximal bundle algorithm for nonconvex inexact functions presented in [16] and contains some closer analysis of the method. In chapter 6 a variable metric variant of that algorithm is developed using the nonsmooth second model suggested in [47] and [45]. This method makes it possible to incorporate second order information into the algorithm in order to speed up convergence. The two methods are compared on different academic examples. At last the bundle method is used on the application of parameter optimization in support vector classification in chapter 7.

2. Preliminaries

When it comes to nonsmooth objective functions the derivative based framework of nonlinear optimization methods does not work any more. Meanwhile there exists a well understood theory of 'subdifferential calculus' that gives similar results in the nondifferentiable case. The most important definitions and results of this theory together with some remarks on notation are stated in this chapter.

2.1. Notation

Let x denote a column vector. The transpose of x is denoted by $x^\top$. The scalar product is written $\langle \cdot, \cdot \rangle$. In this thesis generally the euclidean norm is used and denoted by $\| \cdot \|$. In chapter 6 additionally a norm is used that is induced by a symmetric matrix. Here we use the notation $\|x\|_A^2 = \langle x, Ax \rangle$. Inequalities written for vectors $x^1 \leq x^2$, $x^1, x^2 \in \mathbb{R}^n$ are to be read component wise. With 0 we denote the zero vector of appropriate size. The identity matrix of appropriate size is written as $\mathbb{I}$.

As we work with numerical methods in this thesis occur a lot of sequences of various dimensions. For vectors iteration indices are indicated by a superscript x^k whereas the components are indicated by subscripts $x = (x_1, x_2, ..., x_n)^\top$. Sequences of numbers and matrices a indexed with subscripts. For (sub-)sequences where k comes from an index set $K \subset \mathbb{N}$ we write $\{x^k\}_{k \in K}$. If k is in the natural numbers this notation is shortened to $\{x^k\}$. We denote the open ball around x with radius r with $B_r(x)$. The subset relation is denoted by $A \subset B$. It is to be read in the sense that A is a subset of B or that $A = B$.

2.2. Nonsmooth Analysis

Throughout this thesis we consider different optimization problems of the form

$$\min_x f(x) \quad \text{s.t.} \quad x \in X \subset \mathbb{R}^n$$

where f is a possibly nonsmooth function.

Nonsmooth functions have kinks where a unique gradient cannot be defined. It is however possible to define a set of tangents to the graph called subdifferential. The subdifferential was first defined for convex functions.

Definition 2.1 ([20, Definition 1.2.1, p. 241]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function.

The *subdifferential* of f at $x \in \mathbb{R}^n$ is the set

$$\partial f(x) := \{g \in \mathbb{R}^n \mid f(y) - f(x) \geq \langle g, y - x\rangle \quad \forall y \in \mathbb{R}^n\}.$$

The subdifferential is a set valued mapping. It is closed and convex. If f is differentiable, its subdifferential is single valued and coincides with its gradient $\partial f(x) = \nabla f(x)$ [50].

It is also possible to define a subdifferential for nonconvex functions. This is the subdifferential we will work with in this thesis most of the time.

Definition 2.2 (c.f. [4, p. 25, 27]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz (and not necessarily convex). The *subdifferential* of f at $x \in \mathbb{R}^n$ is the set

$$\partial f(x) := \left\{ g \in \mathbb{R}^n \;\middle|\; \limsup_{y \to x,\ h \searrow 0} \frac{f(y + hv) - f(y)}{h} \geq \langle g, v\rangle \quad \forall v \in \mathbb{R}^n \right\}.$$

All convex functions are locally Lipschitz [20, Theorem 3.1.1, p. 16] so the above definition holds also for convex functions. In fact if the function is convex the subdifferential from definition 2.2 is equivalent to the one from definition 2.1 [4, Proposition 2.2.7, p. 36]. Due to this equivalence we call elements from both subdifferentials subgradients.

Remark: It is important to observe that subgradient inequality

$$f(y) - f(x) \geq \langle g, y - x\rangle \quad \forall y \in \mathbb{R}^n \tag{2.1}$$

only holds in the convex case.

There is also a sum rule for the subdifferential.

Proposition 2.3 ([4, Proposition 2.3.3, p. 38]) Let $F(x) = \sum_i f_i(x)$ be a finite sum of nondifferentiable functions. Then it holds

$$\partial F(x) \subset \sum_i \partial f_i(x).$$

Analogous to the $\mathcal{C}^1$-case some first order optimality conditions can be stated. For nondifferentiable functions a *stationary point* x of the function f is characterized by [4, p. 38]

$$0 \in \partial f(x).$$

If the function f is convex, then every stationary point is a minimum.

A drawback of the subdifferential is that it does not indicate how near the evaluated point is to a stationary point or minimum of a function. This can only be seen if the evaluated point is already stationary.

This issue is addressed by the *ε-subdifferential*. It gathers all information in a small neighborhood of the point x.

For convex functions an *ε-subgradient* of $f(x)$ is defined as a vector $g \in \mathbb{R}^n$ satisfying the inequality

$$f(y) - f(x) \geq \langle g, y - x \rangle - \varepsilon \quad \forall y \in \mathbb{R}^n.$$

The ε-subdifferential is then the set

$$\partial_\varepsilon f(x) := \{g \in \mathbb{R}^n \mid g \text{ is an } \varepsilon\text{-subgradient of } f(x)\}.$$

For nonconvex functions the subdifferential that is used in this thesis is the *Fréchet ε-subdifferential*.

Definition 2.4 (c.f. [22, p. 73]) The Fréchet ε-subdifferential of $f(x)$ is

$$\partial_{[\varepsilon]} f(x) := \left\{ g \in \mathbb{R}^n \;\middle|\; \liminf_{\|h\| \to 0} \frac{f(x+h) - f(x) - \langle g, h \rangle}{\|h\|} \geq -\varepsilon \right\}.$$

For $\varepsilon = 0$ this is called *Fréchet subdifferential*. For convex functions the Fréchet ε-subdifferential and the ε-subdifferential are *not* the same.

In the course of this thesis we sometimes derive stronger results for a smaller class of nonsmooth functions. Those functions are called lower-$\mathcal{C}^2$ functions and can be defined as follows.

Definition 2.5 (c.f. Definition 10.29, p. 447 in [51]) A function $f : \mathbb{R}^n \to \mathbb{R}$ is *lower-$\mathcal{C}^2$*, if on some neighborhood Ω of each $\bar{x} \in \mathbb{R}^n$ there exists a representation

$$f(x) = \max_{t \in T} f_t(x)$$

where the functions f_t are of class $\mathcal{C}^2$ on Ω and $f_t(x)$ and all its first and second partial derivatives depend continuously on both x and $(x, t) \in \Omega \times T$. The index set T is a compact space.

Lower-$\mathcal{C}^2$ functions are locally Lipschitz continuous [51, Theorem 10.31, p. 448].

3. A Basic Bundle Method

When bundle methods were first introduced in 1975 by Lemaréchal and Wolfe they were developed to minimize a convex (possibly nonsmooth) function f for which at least one subgradient at any point x can be computed [39]. To provide an easier understanding of the proximal bundle method from [16] presented in chapter 5 and stress the most important ideas of how to deal with nonconvexity and inexactness in bundle methods first a basic bundle method is shown here.

Bundle methods can be interpreted in two different ways: From the dual point of view one tries to approximate the ε-subdifferential to finally ensure first order optimality conditions. The primal point of view interprets the bundle method as a stabilized form of the cutting plane method where the objective function is modeled by tangent hyperplanes [15]. We focus here on the primal approach.

3.1. Derivation of the Bundle Method

This section gives a short summery of the derivations and results of chapter XV in [19] where a primal bundle method is derived as a stabilized version of the cutting plane method. If not otherwise indicated the results in this section are therefore taken from chapter XV in [19].

The optimization problem considered in this chapter is

$$\min_x f(x) \quad \text{s.t.} \quad x \in X \tag{3.1}$$

where f is a convex but possibly nondifferentiable function and $X \subset \mathbb{R}^n$ is a closed and convex set.

3.1.1. A Stabilized Cutting Plane Method

The geometric idea of the *cutting plane method* is to build a piecewise linear model of the objective function f that can be minimized more easily than the original objective function. This model is built from a *bundle* of information that is gathered in the previous iterations. In the k'th iteration, the bundle consists of the previous iterates x^j, the respective function values $f(x^j)$ and a subgradient at each point $g^j \in \partial f(x^j)$ for all indices j in the index set J_k. From each of these triples one can construct a linear function

$$l_j(x) := f(x^j) + \left\langle g^j, x - x^j \right\rangle$$

where $f(x^j) = l_j(x^j)$ and due to convexity $f(x) \geq l_j(x),\ x \in X$.
The objective function f can then be approximated by the piecewise linear function

$$m_k(x) := \max_{j \in J_k} l_j(x). \tag{3.2}$$

This function is therefore also called *model function*. Instead of working with the original objective, a new iterate x^{k+1} is found by solving the subproblem

$$\min_x m_k(x) \quad \text{s.t.} \quad x \in X.$$

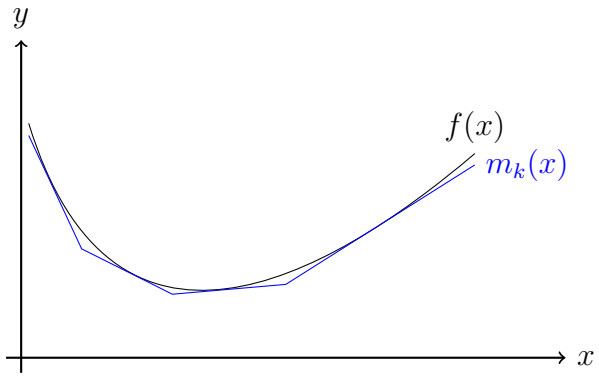

Figure 1: *Cutting plane model m_k of a function f.*

This subproblem should of course be easier to solve than the original task. A question that depends a lot on the structure of X. If $X = \mathbb{R}^n$ or a polyhedron, the problem can be solved easily. Still there are some major drawbacks to the idea. For example if $X = \mathbb{R}^n$ the solution of the subproblem in the first iteration is always $-\infty$. In general we can say that the subproblem does not necessarily have a solution. To tackle this problem a regularization term is introduced to the subproblem. It then reads

$$\min \tilde{m}_k(x) = m_k(x) + \frac{1}{2t_k}\|x - x^k\|^2 \quad \text{s.t.} \quad x \in X,\ t_k > 0. \tag{3.3}$$

This new subproblem is strongly convex and therefore always has a unique solution.

The regularization term can be motivated and interpreted in many different ways (c.f. [19, chapter XV]). From different possible regularization terms the most popular in bundle

methods is the penalty-like regularization used here.

The second major step towards the bundle algorithm is the introduction of a so called *stability center* or *serious point* $\hat{x}^k$. It is the iterate that yields the "best" approximation of the optimal point up to the k'th iteration (not necessarily the lowest function value though). The updating technique for $\hat{x}^k$ is crucial for the convergence of the method: If the next iterate yields a decrease of f that is "large enough", namely larger than a fraction of the decrease suggested by the model function for this iterate, the stability center is moved to that iterate. If this is not the case, the stability center remains unchanged.

In practice this is implemented as follows: First define the *model decrease* δ_k^M which is the decrease of the model for the new iterate x^{k+1} compared to the function value at the current stability center $\hat{x}^k$

$$\delta_k^M := f(\hat{x}^k) - m_k(x^{k+1}) \geq 0. \tag{3.4}$$

If the actual decrease of the objective function is larger than a fraction of the model decrease

$$f(\hat{x}^k) - f(x^{k+1}) \geq m\delta_k^M, \quad m \in (0,1)$$

set the stability center to $\hat{x}^{k+1} = x^{k+1}$. This is called a *serious* or *descent step*. If this is not the case a *null step* is executed and the serious iterate $\hat{x}^{k+1} = \hat{x}^k$ remains the same.

Beside the model decrease other forms of decrease measures and variations of these are possible. Some are presented in [19] and [8].

3.1.2. Subproblem Reformulations

The subproblem to be solved to find the next iterate can be rewritten as a smooth optimization problem. For convenience we first rewrite the affine functions l_j with respect to the stability center $\hat{x}^k$:

$$\begin{aligned} l_j(x) &= f(x^j) + \left\langle g^j, x - x^j \right\rangle \\ &= f(\hat{x}^k) + \left\langle g^j, x - \hat{x}^k \right\rangle - \left(f(\hat{x}^k) - f(x^j) + \left\langle g^j, x^j - \hat{x}^k \right\rangle \right) \\ &= f(\hat{x}^k) + \left\langle g^j, x - \hat{x}^k \right\rangle - e_j^k \end{aligned}$$

where

$$e_j^k := f(\hat{x}^k) - f(x^j) + \left\langle g^j, x^j - \hat{x}^k \right\rangle \geq 0 \quad \forall j \in J_k \tag{3.5}$$

is the *linearization error*. Due to convexity of f it is nonnegative. This property is essential for the convergence theory and will also be of interest when moving on to the case of nonconvex and inexact objective functions.

Subproblem (3.3) can now be written as

$$\min_{\hat{x}^k + d \in X} \tilde{m}_k(\hat{x}^k + d) = f(\hat{x}^k) + \max_{j \in J_k} \left\{ \left\langle g^j, d \right\rangle - e_j^k \right\} + \frac{1}{2t_k} \|d\|^2 \tag{3.6}$$

$$\Leftrightarrow \quad \min_{\substack{\hat{x}^k + d \in X, \\ \xi \in \mathbb{R}}} \xi + \frac{1}{2t_k} \|d\|^2 \quad \text{s.t.} \quad \left\langle g^j, d \right\rangle - e_j^k - \xi \leq 0, \quad j \in J_k \tag{3.7}$$

where $d := x - \hat{x}^k$ and the constant term $f(\hat{x}^k)$ was discarded for the sake of simplicity. If X is a polyhedron this is a convex quadratic optimization problem that can be solved using standard methods of nonlinear optimization. It should however be observed that the matrix of the quadratic part is only positive semidefinite because it does not have full rank.

The pair (ξ_k, d^k) solves (3.7) if and only if

$$\begin{aligned} &d^k \text{ solves the original subproblem (3.6) and} \\ &\xi_k = \max_{j \in J_k} {g^j}^\top d^k - e_j^k = m_k(\hat{x}^k + d^k) - f(\hat{x}^k). \end{aligned} \tag{3.8}$$

The new iterate is given by $x^{k+1} = \hat{x}^k + d^k$.

3.2. The Prox-Operator

The constraint $\hat{x}^k + d \in X$ can also be incorporated directly in the objective function by using the *indicator function*

$$\mathtt{i}_X(x) := \begin{cases} 0, & \text{if } x \in X \\ +\infty, & \text{if } x \notin X \end{cases}. \tag{3.9}$$

This function is convex if and only if the set X is convex [51, p. 40].

Remark: The indicator function is actually an extended-real-valued function, meaning that it allows the function value $+\infty$. Introducing it into the subproblem means that the objective function of the subproblem also becomes an extended-real-valued function. As this is does not have any impact on the convergence theory we omit to introduce the concept of extended-real-valued functions here.

Subproblem (3.3) then reads with respect to the serious point $\hat{x}^k$

$$\min_{x\in\mathbb{R}^n} m_k(x) + \mathtt{i}_X(x) + \frac{1}{2t_k}\|x-\hat{x}^k\|^2. \tag{3.10}$$

The subproblem is now written as the *Moreau-Yosida regularization* of $\check{f}(x) := m_k(x) + \mathtt{i}_X(x)$. The emerging mapping is also known as *proximal point mapping* [15] or *prox-operator*

$$prox_{t,\check{f}}(x) := \operatorname*{arg\,min}_{y\in\mathbb{R}^n}\left\{\check{f}(y) + \frac{1}{2t}\|x-y\|^2\right\}, \quad t>0. \tag{3.11}$$

This special form of the subproblems gives the primal bundle method its name, *proximal bundle method*. The above mapping also plays a key role when the method is generalized to nonconvex objective functions and inexact information.

3.3. Aggregation and Stopping Condition

We look again at a slightly different formulation of the bundle subproblem

$$\begin{aligned}\min_{\substack{d\in\mathbb{R}^n,\\ \xi\in\mathbb{R}}} \quad & \xi + \mathtt{i}_X(\hat{x}^k+d) + \frac{1}{2t_k}\|d\|^2\\ \text{s.t.} \quad & \left\langle g^j, d\right\rangle - e_j^k - \xi \le 0, \quad j\in J_k.\end{aligned}$$

As the objective function is still convex (X is a convex set) the following Karush-Kuhn-Tucker (KKT) conditions have to be valid for the minimizer $\left(\xi_k, d^k\right)$ of the above subproblem [20] assuming a constraint qualification holds if the constraint set X makes it necessary.

There exist a subgradient $\nu^k \in \partial\mathtt{i}_X(\hat{x}^k+d^k)$ and Lagrangian multipliers $\alpha_j,\ j\in J^k$ such that

$$0 = \nu^k + \frac{1}{t_k} d^k + \sum_{j \in J^k} \alpha_j g^j, \tag{3.12}$$

$$\sum_{j \in J_k} \alpha_j = 1, \tag{3.13}$$

$$\alpha_j \geq 0, \ j \in J^k, \tag{3.14}$$

$$\left\langle g^j, d^k \right\rangle - e_j^k - \xi_k \leq 0 \text{ and} \tag{3.15}$$

$$\sum_{j \in J^k} \alpha_j \left(\left\langle g^j, d^k \right\rangle - e_j^k - \xi_k \right) = 0. \tag{3.16}$$

From condition (3.12) follows that

$$d^k = -t_k \left(G^k + \nu^k \right) \tag{3.17}$$

with the *aggregate subgradient*

$$G^k := \sum_{j \in J^k} \alpha_j g^j \quad \in \partial m_k(x^{k+1}). \tag{3.18}$$

The fact that G^k belongs to the subdifferential of the k'th model m_k at the point $\hat{x}^k + d^k$ follows from noting that

$$0 \in \partial m_k(\hat{x}^k + d^k) + \partial \mathtt{i}_X(\hat{x}^k + d^k) + \frac{1}{2t_k} d^k$$

is the optimality condition derived from formulation (3.10) by the sum rule for subdifferentials and comparing the different components with the ones derived in (3.12).

Rewriting condition (3.16) yields the *aggregate error*

$$E_k := \sum_{j \in J^k} \alpha_j e_j^k = \left\langle G^k, d^k \right\rangle + f(\hat{x}^k) - m_k(x^{k+1}). \tag{3.19}$$

Here relation (3.8) was used to replace ξ_k.

The aggregate subgradient and error are used to formulate an implementable stopping condition for the bundle algorithm. The motivation behind that becomes clear with the following lemma.

Lemma 3.1 ([11, Theorem 6.68, p. 387]) *Let $X = \mathbb{R}^n$. Let $\varepsilon > 0$, $\hat{x}^k \in \mathbb{R}^n$ and*

$g^j \in \partial f(x^j)$ *for* $j \in J^k$. *Then the set*

$$\mathcal{G}_\varepsilon^k := \left\{ \sum_{j \in J^k} \alpha_j g^j \;\middle|\; \sum_{j \in J^k} \alpha_j e_j \leq \varepsilon, \sum_{j \in J^k} \alpha_j = 1, \alpha_j \geq 0, j \in J^k \right\}$$

is a subset of the ε*-subdifferential of* $f(\hat{x}^k)$

$$\mathcal{G}_\varepsilon^k \subset \partial_\varepsilon f(\hat{x}^k).$$

This means that in the unconstrained case $G^k \in \partial_{E_k} f(\hat{x}^k)$. So driving $\|G^k\|$ and E_k to zero results in some approximate ε-optimality of the objective function. In the constrained case the stopping condition is written as

$$\delta_k = E^k + t_k \|G^k + \nu^k\|^2 \leq \texttt{tol},$$

for a fixed tolerance $\texttt{tol} > 0$.

The decrease measure δ_k is also taken for the decrease test. The relation

$$\begin{aligned} \delta_k &= E^k + t_k \|G^k + \nu^k\|^2 \\ &= E^k - \left\langle G^k, d^k \right\rangle - \left\langle \nu^k, d^k \right\rangle \\ &= f(\hat{x}^k) - m_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle, \end{aligned}$$

where (3.18) and (3.19) were used, shows that the new δ_k is only a small variation of the model decrease δ_k^M. If the iterate x^{k+1} does not lie on the boundary of the constraint set X, the vector ν^k is equal to zero and the expression simplifies to the one stated in (3.4).

For the model update the following two conditions are assumed to be fulfilled in consecutive null steps:

$$m_{k+1}(\hat{x}^k + d) \geq f(\hat{x}^{k+1}) - e_{k+1}^{k+1} + \left\langle g^{k+1}, d \right\rangle \quad \forall d \in \mathbb{R}^n \text{ and} \tag{3.20}$$

$$m_{k+1}(\hat{x}^k + d) \geq a_k(\hat{x}^k + d) \quad \forall d \in \mathbb{R}^n. \tag{3.21}$$

The first condition means that the newly computed information is always put into the bundle. The second one is important when updating the bundle index set J^k. It holds

trivially if no or only inactive information j with $\alpha_j = 0$ is removed [16]. It is also always satisfied if the aggregate linearization a_k itself is added to the bundle. In this case active information can be removed without violating the condition. This is the key idea of Kiwiel's aggregation technique and ensures that the set $\{j \in J^k \mid \alpha_j > 0\}$ can be bounded also for nonpolyhedral constraint sets (c.f. [16, Remark 2, p. 11]).

An issue of bundle methods is that in spite of the possibility to delete inactive information the bundle can still become very large. Kiwiel therefore proposed a totally different use of the aggregate objects in [24]. The aggregate subgradient can be used to build the *aggregate linearization*

$$a_k(\hat{x}^k + d) := m_k(x^{k+1}) + \langle G^k, d - d^k \rangle.$$

This function can be used to avoid memory overflow as it compresses the information of all bundle elements into one affine plane. Adding the function a_k to the cutting plane model preserves the assumptions (3.20) and (3.21) put on the model and can therefore be used instead of or in combination with the usual cutting planes.

This can however impair the speed of convergence if the bundle is kept too small and provides hence less information about the objective function [7, p. 654].

3.4. The Algorithm

We have now all the ingredients so that the following basic bundle algorithm can be stated:

Algorithm 3.1: Basic Bundle Method

Select a descent parameter $m \in (0, 1)$ and a stopping tolerance $\texttt{tol} \geq 0$. Choose a starting point $x^1 \in \mathbb{R}^n$ and compute $f(x^1)$ and g^1. Set the initial index set $J_1 := \{1\}$ and the initial stability center to $\hat{x}^1 := x^1$. Set $f(\hat{x}^1) = f(x^1)$ and select $t_1 > 0$. Initialize $m_1(\hat{x}^1 + d) = f(\hat{x}^1) + \langle g^1, d \rangle$ and $e_1^1 = 0$.
For $k = 1, 2, 3 \ldots$

1. Calculate
$$d^k = \operatorname*{arg\,min}_{d \in \mathbb{R}^n} m_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2t_k}\|d\|^2$$
and the corresponding Lagrange multipliers $\alpha_j^k,\ j \in J_k$.

2. Set

$$G^k = \sum_{j \in J_k} \alpha_j^k g_j^k,$$
$$E_k = \sum_{j \in J_k} \alpha_j^k e_j^k \text{ and}$$
$$\delta_k = E_k + \frac{1}{t_k} d_k^2.$$

   If $\delta_k \leq \texttt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.

4. Compute $f(x^{k+1}),\ g^{k+1}$.
   If $f(x^{k+1}) \leq f(\hat{x}^k) - m\delta_k \quad \rightarrow$ serious step:
   Set $\hat{x}^{k+1} = x^{k+1},\ f(\hat{x}^{k+1}) = f(x^{k+1})$ and select a suitable $t_{k+1} > 0$.
   Otherwise $\quad \rightarrow$ nullstep:
   Set $\hat{x}^{k+1} = \hat{x}^k,\ f(\hat{x}^{k+1}) = f(\hat{x}^k)$ and choose $t_{k+1} > 0$ in a suitable way.

5. Select the new bundle index set J_{k+1}, calculate e_j^{k+1} by (3.5) for all $j \in J_{k+1}$ and update the model m_{k+1}.

In steps 4 and 5 of the algorithm it is not specified how to update the parameter t_k, the index set J^k and the model m_k. For the convergence proof it is only necessary that $\liminf_{k\to\infty} t_k > 0$ and that conditions (3.20) and (3.21) are fulfilled.

In practice the choice of t_k can be realized by taking

$$t_{k+1} = \kappa_+ t_k, \quad \kappa_+ > 1 \tag{3.22}$$

at every serious step and

$$t_{k+1} = \max\{\kappa_- t_k, t_{min}\}, \quad \kappa_- < 1 \text{ and } t_{min} > 0 \tag{3.23}$$

at every null step. The idea behind this management of t_k is taken from the trust region method: If the computed iterate was good, the model is assumed to be reliable in a larger area around this serious iterate so bigger step sizes are allowed. If a null step was taken, the model seems to be too inaccurate far from the current serious point. Then smaller step sizes are used. A more sophisticated version of this kind of step size management is also used by Noll et al. in [47] and [45]. The trust region idea was very much exploited by

Schramm and Zowe in [52]. In the case $X = \mathbb{R}^n$ the sequence $\{\hat{x}^k\}$ can be unbounded. In this case bounding $t_k \leq t_{max} < \infty$ for all k preserves the convergence proof [19, Theorem 3.2.2, p. 308].

In general it can be shown that if f possesses global minima and the basic bundle algorithm generates the sequence $\{\hat{x}^k\}$, this sequence converges to a minimizer of problem (3.1) (c.f [19]).

4. Variations of the Bundle Method

After their discovery in 1975 bundle methods soon became very successful. Only a few years later they were generalized to be used also with nonconvex objective functions. Early works that contain fundamental ideas still used for these algorithms are [38] and [23]. It then took over 25 years that bundle methods were again generalized to the use of inexact information, first works on this subject being [18, 25] and [54].

This chapter of the thesis shortly presents the key ideas of those two kinds of generalizations and different types of bundle methods that realize them. This is first done for the case of convex objective functions with inexact function values and/or subgradient information and then for nonconvex objective functions.

4.1. Convex Bundle Methods with Inexact Information

We focus here on *convex* bundle methods with inexact information. The reason for this is that there is a fundamental difference in treating inexactness between methods that assume convex and those that assume nonconvex objective functions. When dealing with nonconvex objective functions inexactness is treated as some additional nonconvexity therefore no additional strategies are used to cope with the noise. This is not possible if the convexity property is to be exploited for better convergence results. A thorough study on this subject including a synthetic convergence theory is done in [8]. Here the most important aspects of that paper are reviewed.

4.1.1. Different Types of Inexactness

Throughout this chapter we consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{4.1}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$ is a finite convex function. The function values and one subgradient at each point x are given by an inexact oracle. It is reasonable to define different kinds of inexactness and further assumptions can be put on the noise to achieve stronger convergence results. However, generally inexact information for convex objective functions is defined in the following way:

$$f_x := f(x) - \sigma_x, \quad \sigma_x \leq \bar{\sigma} \text{ and} \tag{4.2}$$

$$g_x \in \mathbb{R}^n \text{ such that } f(\cdot) \geq f_x + \langle g_x, \cdot - x \rangle - \theta_x, \quad \theta_x \leq \bar{\theta}. \tag{4.3}$$

From this follows because of

$$f(\cdot) \geq f(x) + \langle g_x, \cdot - x \rangle - (\sigma_x + \theta_x) \tag{4.4}$$

that g_x is an ε-subgradient of $f(x)$ with $\varepsilon := \sigma_x + \theta_x \geq 0$ independently of the signs of the errors.

Different convergence results for the applied bundle methods are possible depending on if the bounds $\bar{\sigma}$ and $\bar{\theta}$ are unknown, known or even controllable.

In case of controllability of $\bar{\sigma}$ and $\bar{\theta}$ it may be possible to drive them to zero as the iterations increase such that $\lim_{k\to\infty} \sigma_k = 0$ and $\lim_{k\to\infty} \theta_k = 0$. We talk then of *asymptotically vanishing errors*. This case is important because it allows convergence to the exact minimum of the problem even if function values and subgradients are erroneous. In the case of $\bar{\theta} = 0$ it even suffices to show that the errors are only asymptotically exact for descent steps [26]. This observation was the motivation for the partly inexact bundle methods presented in [26] and [8]. The idea is to calculate a value of the objective function with a demanded accuracy (which is finally going to be exact) only if a certain target descent γ_x is reached. This approach can save a lot of (unnecessary) computational effort while still enabling convergence to the exact minimum (c.f. [8]).

In view of good convergence properties oracles that only underestimate the true function, so called *lower oracles*, are also very interesting. Lower oracles provide f_x and g_x such that $f_x \leq f(x)$ and $f(\cdot) \geq f_x + \langle g_x, \cdot - x \rangle$. That means the cutting plane model is always minorizing the true function as it is the case for exact information. In this case if the value to approximate the optimal function value is chosen properly, it is not necessary to include any new steps into the method to cope with the inexactness, such as noise attenuation [8, Corollary 5.2, p. 256].

4.1.2. Noise Attenuation

In the case of inexact information, especially if the inexact function value can overestimate the real one, it is possible that the aggregate linearization error E_k becomes very small (or

even negative) even though the current iterate is far from the minimum of the objective function. To tackle this problem the authors propose a procedure called *noise attenuation* that was developed in [18] and [25]. The basic idea is to allow bigger step sizes t_k whenever the algorithm comes in the situation described above. This ensures that either some significant descent towards the real minimum can be done or shows that the point where the algorithm is stuck is actually such a minimum. Noise attenuation is triggered when E_k or respectively the descent measure δ_k that is used for the descent test is negative. A more detailed description is given in [8].

4.1.3. Convergence Results

Depending on the kind of error many slightly different convergence results can be proven for bundle methods that handle convex objective functions with inexact information. In case of the general error defined in (4.2) and (4.3) it can be shown that for bounded sequences $\{\hat{x}^k\}$ every accumulation point $\bar{x}$ of an infinite series of serious steps or the last serious iterate before an infinite tail of null steps is a $\bar{\sigma}$-solution of the problem meaning that

$$f(\bar{x}) \leq f^* + \bar{\sigma}$$

with f^* being an exact solution of problem (4.1).

Generally for asymptotically vanishing errors it is possible to construct bundle methods very similar to the basic bundle method that converge to the exact minimum of the problem. For more detailed results refer to [8].

4.2. Nonconvex Bundle Methods with Exact Information

In the nonconvex case the optimization problem is the following:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{4.5}$$

This time $f : \mathbb{R}^n \to \mathbb{R}$ is a finite, locally Lipschitz function. It is neither expected to be convex nor differentiable.

In the case of inexactness in convex bundle methods, where a lot of different assumptions can be put on the errors to reach different convergence results, the strategy to cope

with these errors remains very much the same. In contrast to this in case of nonconvex objective functions the set of functions to be studied is rather uniform still there exist very different approaches to tackle the problem. As the nonnegativity property of the linearization errors e_j^k is crucial for the convergence proof of convex bundle methods an early idea was forcing the errors to be so by different downshifting strategies. A very common one is using the *subgradient locality measure* [24, 38]. Here the linearization error is essentially replaced by the nonnegative number

$$\tilde{e}_j^k := \max_{j \in J_k} \{|e_j^k|, \gamma \|\hat{x}^k - x^j\|^2\}$$

or a variation of this expression.

The expression gradient locality measure comes from the dual point of view, where the aggregate linearization error provides a measure for the distance of the calculated ε-subgradient to the objective function.

Methods that use downshifting for building the model function are often endowed with a line search to provide sufficient decrease of the objective function. For the linesearch to terminate finitely usually semismoothness of the objective function is needed.

4.2.1. Proximity Control

Instead of using line search it is also possible to do *proximity control*. This means that the step size parameter t_k is managed in a smart way to ensure the right amount of decrease in the objective function. This method is very helpful in the case of nonconvex objective functions with inexact information as it is predominantly considered in this thesis.

As inexactness can be seen as a kind of slight nonconvexity one could be tempted to think that nonconvex bundle methods are destined to be extended to the inexact case. Indeed, the two existing algorithms [16] and [45] that deal with both nonconvexity and inexactness are both extensions of a nonsmooth bundle method. This is however seldom possible for algorithms that employ a line search because for functions with inexact information convergence of this subroutine cannot be proven.

To this end proximity control seems to be a very promising strategy. It is used in many different variations in [1, 33, 44, 46, 47] and [52].

4.2.2. Other Concepts

In the beginning bundle methods were mostly explored from the dual point of view. Newer concepts focus also on the primal version of the method. This invokes for example having different model functions for the subproblem.

In [9] and [10] the difference function

$$h(d) := f(x^j + d) - f(x^j), \quad j \in J_k$$

is approximated to find a descent direction of f. The negative linearization errors are addressed by using two different bundles. One contains the indices with nonnegative linearization errors and one contains the other ones. From these two bundles two cutting plane approximations can be constructed which provide the bases for the calculation of new iterates.

In [47] Noll et al. follow an approach of approximating a local model of the objective function. The model can be seen as a nonsmooth generalization of the Taylor expansion and looks the following:

$$\Phi(y, x) = \phi(y, x) + \frac{1}{2}(y - x)^\top Q(x)(y - x).$$

The so called *first order model* $\phi(., x)$ is convex but possibly nonsmooth and can be approximated by cutting planes. The *second order part* is quadratic but not necessarily convex. The algorithm proceeds similarly to a general bundle algorithm. Instead of a line search it uses proximity control to ensure convergence.

Generally for all of these methods convergence to a stationary point is established under the assumptions of a locally Lipschitz objective function and bounded level sets $\{x \in \mathbb{R}^n \mid f(x) \leq f(\hat{x}^1)\}$. If the method uses a line search additionally semismoothness of the objective function is needed.

In [45] the second order approach of [47] is extended to functions with inexact information. As far as we know this is the only other bundle method that can deal with nonconvexity and inexactness in both the function value and subgradient. In this method a lower-$\mathcal{C}^1$ objective function and some assumptions on the form of inexactness are needed to prove convergence.

The above algorithm inspires the variable metric variation of the method used by Hare et al. in [16] that is presented in chapter 6 of this thesis.

5. Proximal Bundle Method for Nonconvex Functions with Inexact Information

This chapter focuses on the proximal bundle method presented by Hare et al. in [16]. The idea is to extend the basic bundle algorithm for nonconvex functions with both inexact function and subgradient information. The key idea of the algorithm is the one already developed by Hare and Sagastizábal in [15]: When dealing with nonconvex functions a very critical difference to the convex case is that the linearization errors are not necessarily nonnegative any more. To tackle this problem the errors are manipulated to enforce nonnegativity. In this case this is done my modeling not the objective function directly but a convexified version of it.

5.1. Derivation of the Method

Throughout this chapter we consider the optimization problem

$$\min_x f(x) \quad \text{s.t.} \quad x \in X. \tag{5.1}$$

The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz and (subdifferentially) regular. $X \subset \mathbb{R}^n$ is assumed to be a convex compact set.

Definition 5.1 [51, Theorem 7.25, p. 260] $f : \mathbb{R}^n \to \mathbb{R}$ is called *subdifferentially regular* at $\bar{x} \in \mathbb{R}^n$ if the epigraph

$$epi(f) := \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid \alpha \geq f(x)\}$$

is Clarke regular at $\bar{x}, f(\bar{x})$.

Clarke regularity basicly means, that epigraph of a function f does not have any 'inward' corners. An exact definition is given in [51, Definition 6.4, p. 199].

5.1.1. Inexactness

It is assumed that both the function value as well as one element of the subdifferential can be provided in an inexact form. For the function value inexactness is defined straight forwardly: If

$$\|f_x - f(x)\| \leq \sigma_x,$$

then f_x approximates the value $f(x)$ within σ_x. This is slightly different from the definition in (4.2). In the convex case it follows from (4.4) that $\bar{\sigma} \geq \sigma_x \geq -\theta_x \geq -\bar{\theta}$ and therefore $f_x \in [f(x) - \bar{\theta}, f(x) + \bar{\sigma}]$ for the overall error bounds $\bar{\sigma}$ and $\bar{\theta}$.

As the 'normal' ε-subdifferential is not defined for nonconvex functions we adopt the notation used in [45] and interpret inexactness in the following way: $g \in \mathbb{R}^n$ approximates a subgradient of $\partial f(x)$ within $\theta \geq 0$ if

$$g_x \in \partial f(x) + B_\theta(0) =: \partial_{[\theta]} f(x)$$

where $\partial f(x)$ is the Clarke subdifferential of f.

The given definition of the inexactness can be motivated by the relation

$$g_x \in \partial_{[\theta]} f(x) \Leftrightarrow g_x \in \partial(f + \theta \| \cdot - x \|)(x)$$

noticed in [58]. It means that the approximation of the subgradient of $f(x)$ is an exact subgradient of a small perturbation of f at x. The set $\partial_{[\varepsilon]} f(x)$ is also known as the Fréchet ε-subdifferential of $f(x)$.

Remark: For convex objective functions this approximate subdifferential does *not* equal the usual convex ε-subdifferential. The two can however be related via

$$\partial_\theta f(x) \subset \partial_{[\theta']} f(x)$$

for a suitable θ'. Generally an explicit relation between θ and θ' is hard to find [45, p. 558].

Like in the paper it is assumed that the errors are bounded although the bound does not have to be known:

$$|\sigma_j| \leq \bar{\sigma}, \bar{\sigma} > 0 \quad \text{and} \quad 0 \leq \theta_j \leq \bar{\theta} \quad \forall j \in J^k \text{ and } \forall k. \tag{5.2}$$

For ease of notation we write from now on f_j and g_j instead of f_{x^j} and g_{x^j} for the approximation of the function value and subgradient at the j'th iterate in the bundle J^k. The approximation at the k'th stability center reads $\hat{f}_k$.

Remark: Before the exact subgradient was denoted with g_j. Here the same notation is used for the approximate subgradient. We leave this double notation for the sake of

readability and remark that for the remainder of this thesis g_j denotes the approximate subgradient. In the few situations, where the exact gradient is needed, it is marked clearly.

5.1.2. Nonconvexity

A main issue both nonconvexity and inexactness entail is that the linearization errors e_j^k are not necessarily nonnegative any more. So based on the results in [14] not the objective function but a convexified version of it is modeled as the objective function of the subproblem.

As already pointed out in section 3.2 the bundle subproblem can be formulated by means of the prox-operator (3.11).

The idea is to use the relation

$$prox_{T=\frac{1}{\eta}+t,f}(x) = prox_{t,f+\eta/2\|\cdot-x\|^2}(x).$$

This means, that the proximal point of the function f for the parameter $T = \frac{1}{\eta} + t$, $\eta, t > 0$, is the same as the one of the convexified function

$$\tilde{f}(y) = f(y) + \frac{\eta}{2}\|y - x\|^2 \tag{5.3}$$

with respect to the parameter t [15]. The parameter η is therefore also called the *convexification parameter* and t the *prox-parameter*.

The main difference of the method in [16] to the basic bundle algorithm 3.1 is that the function that is modeled by the cutting plane model is no longer the original objective function f but the convexified version $\tilde{f}$. This results in the following changes:

In addition to downshifting the linear functions forming the model they have a tilted slope. This is because instead of subgradients of the original objective f subgradients of the function $\tilde{f}$ are taken. We call them *augmented subgradients*. At the iterate x^j such a subgradient is given by

$$s_j^k := g^j + \eta_k \left(x^j - \hat{x}^k\right). \tag{5.4}$$

Downshifting is done in a way that keeps the linearization error nonnegative. The *augmented linearization error* is therefore defined as

$$0 \leq c_j^k := e_j^k + b_j^k, \quad \text{with} \quad \begin{cases} e_j^k := \hat{f}_k - f_j - \left\langle g^j, \hat{x}^k - x^j \right\rangle \\ b_j^k := \frac{\eta_k}{2} \|x^j - \hat{x}^k\|^2 \end{cases} \tag{5.5}$$

and

$$\eta_k \geq \max \left\{ \max_{j \in J_k, x^j \neq \hat{x}^k} \frac{-2e_j^k}{\|x^j - \hat{x}^k\|^2}, 0 \right\} + \gamma.$$

The parameter $\gamma \geq 0$ is a safeguarding parameter to keep the calculations numerically stable.

The new model function can therefore be written as

$$M_k(\hat{x}^k + d) := \hat{f}_k + \max_{j \in J_k} \left\{ \left\langle s_j^k, d \right\rangle - c_j^k \right\}. \tag{5.6}$$

At the proximal center $\hat{x}^k$ holds $M_k(\hat{x}^k) = \hat{f}_k$ for all k by the fact that then $d = 0$ and $c_j^k = 0$.

5.1.3. Aggregate Objects

The definitions of the *augmented aggregate subgradient* S^k, *error* C_k and *linearization* A_k follow straightforwardly from the KKT-conditions. Again $\alpha_j^k, j \in J^k$ denote the Lagrangian multipliers of the subproblem.

$$S^k := \sum_{j \in J_k} \alpha_j^k s_j^k, \tag{5.7}$$

$$C_k := \sum_{j \in J_k} \alpha_j^k c_j^k \text{ and} \tag{5.8}$$

$$A_k(\hat{x}^k + d) := M_k(x^{k+1}) + \left\langle S^k, d - d^k \right\rangle. \tag{5.9}$$

The model decrease is

$$\delta^k := C_k + t_k \|S^k + \nu^k\|^2 = C_k + \frac{1}{t_k} \|d^k\|^2, \tag{5.10}$$

which contains the normal vector

$$\nu^k \in \partial \mathtt{i}_X(x^{k+1}) \tag{5.11}$$

of the constraint set X.

The second formulation in (5.10) follows from the relation $d^k = -t_k(S^k + \nu^k)$ which itself comes from the KKT-conditions.

By the same argumentation as for (3.19) the KKT conditions also reveal another useful characterization of the augmented aggregate linearization error:

$$C_k = \hat{f}_k - M_k(x^{k+1}) + \left\langle S^k, d^k \right\rangle. \tag{5.12}$$

As the model function M_k is convex even for nonconvex objective functions it is still minorized by the aggregate linearization. It holds

$$A_k(\hat{x}^k + d) \leq M_k(\hat{x}^k + d) \quad \forall d \in \mathbb{R}^n. \tag{5.13}$$

The update of t_k can be done in the same way as described in (3.22) and (3.23) for the basic bundle method. Similarly the methods to update the bundle index set J^k stay valid. The update conditions (3.20) and (3.21) for the model are now written with respect to the augmented aggregate linearization and the approximate function value $\hat{f}_{k+1}$.

$$M_{k+1}(\hat{x}^k + d) \geq \hat{f}_{k+1} - c_{k+1}^{k+1} + \left\langle s^{k+1}, d \right\rangle, \quad \forall d \in \mathbb{R}^n \text{ and} \tag{5.14}$$

$$M_{k+1}(\hat{x}^k + d) \geq A_k(\hat{x}^k + d), \quad \forall d \in \mathbb{R}^n. \tag{5.15}$$

A bundle algorithm that deals with nonconvexity and inexact function and subgradient information can now be stated.

Algorithm 5.1: Nonconvex Proximal Bundle Method with Inexact Information

---

Select parameters $m \in (0,1), \gamma > 0$ and a stopping tolerance $\mathtt{tol} \geq 0$. Choose a starting point $x^1 \in \mathbb{R}^n$ and compute f_1 and g^1. Set the initial index set $J_1 := \{1\}$ and the initial prox-center to $\hat{x}^1 := x^1$. Set $\hat{f}_1 = f_1$, $s_1^1 = g^1$ and select $t_1 > 0$. Initialize $c_1^1 = 0$ and $M_1(\hat{x}^1 + d) = \hat{f}_1 + \langle s_1^1, d \rangle$.

For $k = 1, 2, 3, \ldots$

1. Calculate
$$d^k = \arg\min_{d \in \mathbb{R}^n} \left\{ M_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2t_k}\|d\|^2 \right\}. \tag{5.16}$$

2. Set
$$\begin{aligned} G^k &= \sum_{j \in J_k} \alpha_j^k s_j^k \\ C_k &= \sum_{j \in J_k} \alpha_j^k c_j^k \text{ and} \\ \delta_k &= C_k + \frac{1}{t_k}\|d^k\|^2. \end{aligned}$$
If $\delta_k \leq \mathtt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.

4. Compute f^{k+1}, g^{k+1}.
If $f^{k+1} \leq \hat{f}^k - m\delta_k \quad \rightarrow$ serious step:
Set $\hat{x}^{k+1} = x^{k+1}$, $\hat{f}^{k+1} = f^{k+1}$ and select $t_{k+1} > 0$.
Otherwise $\quad \rightarrow$ nullstep:
Set $\hat{x}^{k+1} = \hat{x}^k$, $\hat{f}^{k+1} = \hat{f}^k$ and choose $0 < t_{k+1} \leq t_k$.

5. Select new bundle index set J_{k+1}, calculate
$$\eta_{k+1} = \max\left\{ \max_{j \in J_{k+1}, x^j \neq \hat{x}^{k+1}} \frac{-2e_j^{k+1}}{|x^j - \hat{x}^{k+1}|^2}, 0 \right\} + \gamma$$
and c_j^{k+1} by (5.5) for all $j \in J^{k+1}$. Update the model M_{k+1} such that conditions (5.14) and (5.15) are fulfilled.

5.2. On Different Convergence Results

In terms of usability of the described algorithm it is interesting to see if stronger convergence results are possible if additional assumptions are put on the objective function. This is investigated in the following section.

5.2.1. The Constraint Set

The constraint set X ensures the boundedness of the sequence $\{\hat{x}^k\}$. This is not necessary if the objective function is assumed to have bounded level sets $\{x \in \mathbb{R}^n \mid f(x) \leq f(\hat{x}^1)\}$, an assumption commonly used when optimizing nonconvex functions. As the objective function is assumed to be continuous bounded level sets are compact. Additionally the descent test ensures that $f(\hat{x}^{k+1}) \leq f(\hat{x}^k)$ for all k. The proof holds therefore in the same way as with the set X.

In [8] another stopping criterion is proposed that ensures convergence even for unbounded sequences $\{\hat{x}^k\}$.

5.2.2. Exact Information and Vanishing Errors

As the presented algorithm was originally designed for nonconvex objective functions where function values as well as subgradients are available in an exact manner, all convergence results stay the same with the error bounds $\bar{\sigma} = \bar{\theta} = 0$. As already indicated previously this is the case because inexactness can be seen as a kind of nonconvexity and no additional concepts had to be added to the method when generalizing it to the inexact setting.

If we additionally require the objective function to be lower-$\mathcal{C}^2$ it can be proven that the sequence $\{\eta_k\}$ is bounded [15, Lemma 3, p. 2454]. This is not possible in the case of inexact information even for convex objective functions (see example in [16, p. 22]).

For asymptotically vanishing errors, meaning $\lim_{k\to\infty} \sigma_k = 0$ and $\lim_{k\to\infty} \theta_k = 0$ the convergence theory holds equally well with error bounds $\bar{\sigma} = \bar{\theta} = 0$ in [16, Lemma 5, p. 11]. Still it is difficult if not impossible to show that the sequence $\{\eta_k\}$ is bounded without further assumptions. Under the assumption that f is lower-$\mathcal{C}^2$ and some continuity bounds on the errors

$$\frac{|\sigma_j - \hat{\sigma}_k|}{\|x^j - \hat{x}^k\|^2} \leq L_\sigma, \qquad \frac{\theta_j}{\|x^j - \hat{x}^k\|} \leq L_\theta \quad \forall k \text{ and } \forall j \in J_k$$

boundedness of the sequence $\{\eta_k\}$ can be shown. The question remains however if those assumptions are possible to be assured in practice.

5.2.3. Convex Objective Functions

An obvious gain when working with convex objective functions is that the approximate stationarity condition of [16, Lemma 5 (iii), p. 12] is then an approximate optimality condition. If one takes the error definitions (4.2) and (4.3) that are available in the convex case and assumes $X = \mathbb{R}^n$, statement (22) in [16, p. 12] therefore means that

$$0 \in \partial_{\bar{\sigma}+\bar{\theta}} f(\bar{x}).$$

Thus $\bar{x}$ is $(\bar{\sigma} + \bar{\theta})$-optimal.

This follows from the definition of S^k in (5.7) and local Lipschitz continuity of the ε-subdifferential [51, Proposition 12.68, p. 573].

To conclude this section we can say: At the moment there exist two fundamentally different approaches to tackle inexactness in various bundle methods depending on if the method is developed for convex or nonconvex objective functions. In the nonconvex case inexactness is only considered in the paper by Hare, Sagastizàbal and Sodolov [16] presented above and Noll [45]. In these cases the inexactness can be seen as an additional nonconvexity. In practice this means that the algorithm can be taken from the nonconvex case with no or only minor changes. This includes that all results of the exact case remain true as soon as function and subgradient are evaluated in an exact way. In case of convex objective functions with inexact information stronger convergence results are possible. However to be able to exploit convexity in order to achieve those results the algorithms look different from those designed for nonconvex objective functions and are generally not able to deal with such functions.

5.2.4. Aggregation

In [16] it is assumed, that the set $\{j \in J^k \mid \alpha_j^k > 0\}$ is uniformly bounded for all k. In Remark 2 in [16, p. 11] it is mentioned that this is possible if the set X is polyhedral and an active-set solver is used to solve the subproblem.

Another possibility to assure boundedness of the set, is the aggregation technique mentioned in section 3.3. In this context is has however to be mentioned, that it is not immediately clear if the proof of Lemma 5 in [16, p. 11] still holds in that case. For exact function and subgradient information convergence, of a very similar method to algorithm 5.1 is proven also using of the aggregation technique in [15].

The lemma states the following results:

Lemma 5.2 ([16, Lemma 5, p. 11]) *Suppose that the cardinality of the set* $\{j \in J^k \mid \alpha_j^k > 0\}$ *is uniformly bounded in* k.

(i) If $C^k \to 0$ *as* $k \to \infty$, *then*

$$\sum_{j \in J^k} \alpha_j^k \|x^j - \hat{x}^k\| \to 0 \text{ as } k \to \infty.$$

(ii) If additionally for some subset $\tilde{K} \subset \{1, 2, \dots\}$,

$$\hat{x}^k \to \bar{x}, S^k \to \bar{S} \text{ as } K \ni k \to \infty, \text{ with } \{\eta_k \mid k \in \tilde{K}\} \text{ bounded,}$$

then we also have

$$\bar{S} \in \partial f(\bar{x}) + B_{\bar{\theta}}(0).$$

(iii) If in addition $S^k + \nu^k \to 0$ *as* $\tilde{K} \in k \to \infty$, *then* $\bar{x}$ *satisfies the approximate stationarity condition*

$$0 \in (\partial f(\bar{x}) + \partial \mathtt{i}_X(\bar{x})) + B_{\bar{\theta}}(0). \tag{5.17}$$

(iv) Finally if f *is also lower-*$\mathcal{C}^1$, *then for each* $\varepsilon > 0$ *there exists* $\rho > 0$ *such that*

$$f(y) \geq f(\bar{x}) - (\bar{\theta} + \varepsilon)\|y - \bar{x}\| - 2\bar{\sigma}, \quad \text{for all } y \in X \cap B_\rho(\bar{x}). \tag{5.18}$$

For algorithm 5.1 it has to be shown that item (ii) of this lemma also holds if there are aggregate subgradients in the bundle. Here only the missing links are sketched to extend the proof to this case. The full proof can be found in [16, p. 12]

Therefore let (as in [16]) p^j denote for each j the orthogonal projection of the approximate subgradient g^j on the closed convex set $\partial f(x^j)$. Then it holds by assumption on the approximate subgradient that $\|g^j - p^j\| \leq \theta_j \leq \bar{\theta}$. Furthermore let S_n^k denote the (augmented) aggregate subgradients in the k'th bundle. Without loss of generality we can assume that there is only one aggregate subgradient in the bundle. (If there are more, this results just in more aggregate terms of the same form and behavior in the convex sum, so it does not alter the argumentation.) The subscript n of S_n^k signals, that already n aggregate subgradients were incorporated in the current aggregate subgradient.

The proof works with induction over the number of aggregate subgradients contained in the current subgradient.

Denote the multiplier of the k'th index set corresponding to the aggregate subgradient α_n^k. The remainder of the index set J^k, corresponding to the approximate subgradients, is denoted by J_g^k.

To begin the induction recall that at iteration k $S_0^k = \sum_{j \in J_g^k} \alpha_j^k (g^j + \eta_k(x^j - \hat{x}^k))$ by (5.4) and (5.7) and that $J_g^k = J^k$ in this case.

From this follows that

$$\begin{aligned} S_1^k &= \sum_{j \in J_g^k} \alpha_j^k \left(g^j + \eta_k (x^j - \hat{x}^k) \right) + \alpha_n^k S_0^k \\ &= \sum_{j \in J_g^k} \alpha_j^k \left(g^j + \eta_k (x^j - \hat{x}^k) \right) + \alpha_n^k \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} \left(g^j + \eta_{k-1} (x^j - \hat{x}^{k-1}) \right) \\ &= \sum_{j \in J_g^k} \alpha_j^k p^j + \sum_{j \in J_g^k} \alpha_j^k (g^j - p^j) + \eta_k \sum_{j \in J_g^k} \alpha_j^k (x^j - \hat{x}^k) \\ &+ \alpha_n^k \left(\sum_{j \in J_g^{k-1}} \alpha_j^{k-1} p^j + \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} (g^j - p^j) + \eta_{k-1} \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} (x^j - \hat{x}^{k-1}) \right) \end{aligned} \tag{5.19}$$

As it is explained in the paper, due to the uniform boundedness of the set J^k it is possible to consider this set as some fixed index set (for example $\{1, ..., N\}$). Unused indices are filled with $\alpha_j^k = 0$. Let then J be the set of all $j \in J^k$ such that $\liminf_{k \to \infty} \alpha_j^k > 0$. From item (i) of Lemma 5.2 follows then that $\|x^j - \hat{x}^k\| \to 0$. Hence also $\|x^j - \bar{x}\| \leq \|x^j - \hat{x}^k\| + \|\hat{x}^k - \bar{x}\| \to 0$. It holds that $p^j \in \partial f(x^j)$ and $x^j \to \bar{x}$ for $j \in J$ and $\{\alpha_j^k\} \to 0$ for $j \notin J$. Due to the boundedness of the subdifferentials (this follows from Lipschitz continuity of the function f and is explained in more detail for example in the proof of Theorem 6.3) passing on a subsequence $K \subset \{1, 2, ...\}$ yields that there exists $\bar{p}$ with $p^j \to \bar{p}$ for $j \in J$ and $k \in K$. The second and fifth term in the third equation of equation 5.19 are in $B_{\bar{\theta}}(0)$ and the third and sixth term go to zero. Then

$$\begin{aligned} \lim_{K \ni k \to \infty} S_1^k &= \lim_{K \ni k \to \infty} \sum_{j \in J_g^k} \alpha_j^k p^j + \sum_{j \in J_g^k} \alpha_j^k (g^j - p^j) + \eta \sum_{j \in J_g^k} \alpha_j^k (x^j - \hat{x}^k) \\ &+ \lim_{K \ni k \to \infty} \alpha_n^k \left(\sum_{j \in J_g^{k-1}} \alpha_j^{k-1} p^j + \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} (g^j - p^j) + \eta_{k-1} \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} (x^j - \hat{x}^{k-1}) \right) \\ &= \lim_{K \ni k \to \infty} \sum_{j \in J_g^k} \alpha_j^k p^j + \alpha_n^k \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} p^j + \sum_{j \in J_g^k} \alpha_j^k (g^j - p^j) + \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} (g^j - p^j) \\ &= \lim_{K \ni k \to \infty} \sum_{j \in J_g^k} \alpha_j^k p^j + \alpha_n^k \sum_{j \in J_g^{k-1}} \alpha_j^{k-1} p^j + B_{\bar{\theta}}(0), \end{aligned}$$

and due to outer semicontinuity of the Clarke subdifferential [51, Proposition 6.6 p. 202] and $\sum_{j\in J^k} \alpha_j^k = 1$ it follows that

$$\lim_{K\ni k\to\infty} \sum_{j\in J_g^k} \alpha_j^k p^j + \alpha_n^k \sum_{j\in J_g^{k-1}} \alpha_j^{k-1} p^j \in \partial f(\bar{x}).$$

This proves assertion (ii) for a bundle containing one aggregate subgradients that only consist of genuine approximate (augmented) subgradients.

Assume now that $\lim_{K\ni k\to\infty} S_n^k \in \partial f(\bar{x}) + B_{\bar{\theta}}(0)$ holds also for bundles containing aggregate subgradients of the form S_n^k. The inductive step gives then:

$$\begin{aligned} S_{n+1}^k &= \sum_{j\in J_g^k} \alpha_j^k \left(g^j + \eta_k(x^j - \hat{x}^k)\right) + \alpha_{n+1}^k \left(\alpha_n^{k-1} S_n^{k-1} + \sum_{j\in J_g^{k-1}} \alpha_j^{k-1} \left(g^j + \eta_{k-1}(x^j - \hat{x}^{k-1})\right)\right) \\ &= \sum_{j\in J_g^k} \alpha_j^k \left(g^j + \eta_k(x^j - \hat{x}^k)\right) + \alpha_{n+1}^k \sum_{j\in J_g^{k-1}} \alpha_j^{k-1} \left(g^j + \eta_{k-1}(x^j - \hat{x}^{k-1})\right) \\ &\quad + \alpha_{n+1}^k \alpha_n^{k-1} S_n^{k-1} \end{aligned} \tag{5.20}$$

With the same argumentation as above it follows that in the limit the first tow terms of the second equation in (5.20) are in $\partial f(\bar{x}) + B_{\bar{\theta}}(0)$. From the induction assumption follows that also the last term is in $\partial f(\bar{x}) + B_{\bar{\theta}}(0)$. This proves item (ii) also for the use of aggregation technique.

6. Variable Metric Bundle Method

A way to extend the proximal bundle method is to use an arbitrary metric $\frac{1}{2}\langle d, W_k d\rangle$ with a symmetric and positive definite matrix W_k instead of the Euclidean metric for the stabilization term $\frac{1}{2t_k}\|d\|^2$. Methods doing so are called *variable metric bundle methods*. This chapter combines the method of Hare et al. presented in chapter 5 with the second order model function used by Noll in [45] to a metric bundle method suitable for nonconvex functions with noise.

The chapter starts by explaining the ideas from [45] used to extend the method presented above. It then gives a convergence proof for the developed method and concludes with an explicit strategy how to update the metric during the steps of the algorithm.

Throughout this chapter we still consider the optimization problem (5.1). We also keep the names and definitions of the objects used in chapter 5.

6.1. Main Ingredients to the Method

As already mentioned in chapter 3 the stabilization term can be interpreted in many different ways. In the context of this chapter we can understand it as a pretty rough approximation of the curvature of the objective function. In this thesis we work with locally Lipschitz continuous functions. Rademacher's theorem [17, Theorem 3.1, p. 18] states that on any open set $U \subset \mathbb{R}^n$ such a function is differentiable at almost every point in U. The set of points where the function is nondifferentiable is of zero Lebesgue measure. This means that between those points curvature information can be present and can be used to speed up convergence.

6.1.1. Variable Metric Bundle Methods

Variable metric bundle methods use an approach that can be motivated by the thoughts stated above. Instead of using the Euclidean norm for the stabilization term $\frac{1}{2t_k}\|d\|^2$ the metric is derived from a symmetric and positive definite matrix W_k. As the name of the method suggests, this matrix can vary over the iterations of the algorithm. The subproblem in the k'th iteration therefore reads

$$\min_{\hat{x}^k+d\in\mathbb{R}^n} M_k(\hat{x}^k+d) + \mathtt{i}_X(\hat{x}^k+d) + \frac{1}{2}\langle d, W_k d\rangle .$$

As explained in [30, chapter XV.4] like (3.10) this is a Moreau-Yosida regularization of the objective function (on the constraint set), so this subproblem is still strictly convex and has a unique solution. It is however harder to solve especially if the matrices W_k are no diagonal matrices [36, p. 594]. In the unconstrained case or for a very simple constraint set the subproblem can be solved by calculating a quasi Newton step. Such a method is presented by Lemaréchal and Sagastizábal in [31] for convex functions. Lukšan and Vlček use an algorithm in those lines in [63] which is adapted to a limited memory setting by Haarala et al. in [13].

A challenging question is how to update the matrices W_k. It is important that the updating strategy preserves positive definiteness of the matrices and that the matrices stay bounded. The updates that are used most often are the symmetric rank 1 formula (SR1 update) and the BFGS (Broyden-Fletcher-Goldfarb-Shanno) update. These updates make it possible to assure the required conditions with only little extra effort even in the nonconvex case. Concrete instances of the updates are given in [63] and [30].

6.1.2. Noll's Second Order Model

In [47] Noll et al. present a proximal bundle method for nonconvex objective functions. An important ingredient to the method is that not the objective function itself is approximated in the subproblem but a quadratic model of it:

$$\Phi(x, \hat{x}) = \phi(x, \hat{x}) + \frac{1}{2} \langle x - \hat{x}, Q(\hat{x})(x - \hat{x}) \rangle \tag{6.1}$$

The first order model $\phi(\cdot, \hat{x})$ is convex and possibly nonsmooth. The second order part $\frac{1}{2} \langle \cdot - \hat{x}, Q(\hat{x})(\cdot - \hat{x}) \rangle$ is quadratic but not necessarily convex.

As the first order part of this model is convex it can be approximated by a cutting plane model just like the objective function in usual convex bundle methods. The subproblem emerging from this approach is

$$\min_{\hat{x}^k + d} m_k(\hat{x}^k + d) + \frac{1}{2} \left\langle d, Q(\hat{x}^k) d \right\rangle + \frac{1}{2t_k} \|d\|^2$$

where m_k is the usual cutting plane model (3.2) for the convex, nonsmooth function ϕ.

The matrix $Q(\hat{x}^k)$ itself does not have to be positive definite. In fact the only conditions put on this matrix are that it is symmetric and that all eigenvalues are uniformly bounded. We adopt the notation in [45] and write

$$Q(\hat{x}^k) := Q_k = Q_k^\top \quad \text{and} \quad -q\mathbb{I} \preccurlyeq Q_k \preccurlyeq q\mathbb{I} \text{ for a } q > 0.$$

The notation $A \preccurlyeq B$ with $A, B \in \mathbb{R}^{n\times n}$ means that the matrix $(B - A)$ is positive semidefinite. The symbol $\prec$ means analogously that $(B - A)$ is positive definite.

As the matrix Q_k is symmetric it can also be pulled into the stabilization term. This means the k'th bundle subproblem can also be written as

$$\min_{\hat{x}^k+d\in X} m_k(\hat{x}^k + d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d\right\rangle. \tag{6.2}$$

During the algorithm it is assured that $W_k = Q_k + \frac{1}{t_k}\mathbb{I}$ is positive definite, so this is a variable metric subproblem.

Instead of the first order model $\phi(\cdot, \hat{x})$ the convexified objective function (5.3) is used. In the subproblem this function is approximated by the augmented cutting plane model M_k given in (5.6). The final subproblem of this variable metric bundle algorithm is then

$$\min_{\hat{x}^k+d\in X} M_k(\hat{x}^k + d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d\right\rangle. \tag{6.3}$$

The decomposition of the stabilization term into a curvature approximation and a proximal term makes is easier to reach two goals at the same time:

One the one hand, curvature of the objective can be approximated only under the conditions of the boundedness and symmetry of Q_k. No positive definiteness has to be ensured for convergence. On the other hand the proximal term can be used in the trust region inspired way to make a line search obsolete. As already mentioned in chapter 4 this is an advantage especially when working with inexact functions where a line search is not useable.

6.1.3. The Descent Measure

Due to the different formulation of subproblem (6.3) the descent measure δ_k has to be adapted in the variable metric bundle method. In the same way as for (3.17) from the optimality condition

$$0 \in \partial M_k(x^{k+1}) + \partial \mathtt{i}_X(x^{k+1}) + \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k$$

it follows that

$$S^k + \nu^k = -\left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k, \tag{6.4}$$

S^k and ν^k being the augmented aggregate subgradient and outer normal defined in (5.7) and (5.11) respectively.

From this the model decrease (5.10) can be calculated using (5.9), (5.12) and (6.4):

$$\begin{aligned}
\delta_k &:= \hat{f}_k - M_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle \\
&= \hat{f}_k - A_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle \\
&= C_k - \left\langle S^k + \nu^k, d^k \right\rangle \\
&= C_k + \left\langle d^k, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \quad \forall k \in \mathbb{N}.
\end{aligned} \tag{6.5}$$

The new δ_k is used in the same way as in algorithm 5.1 for the descent test and stopping conditions.

Because the changes in the algorithm concern only the stabilization and the decrease measure δ_k all other relations that were obtained for the different parts of the model M_k in chapter 5 are still valid.

6.2. The Variable Metric Bundle Algorithm

The variable metric bundle algorithm can now be stated as a variation of algorithm 5.1.

Algorithm 6.1: Nonconvex Variable Metric Bundle Method with Inexact Information

---

Select parameters $m \in (0,1), \gamma > 0, q > 0, 0 < t_{min} < \frac{1}{q}$ and a stopping tolerance $\texttt{tol} \geq 0$. Choose a starting point $x^1 \in \mathbb{R}^n$ and compute f_1 and g^1. Set the initial metric matrix $Q_1 = \mathbb{I}$, the initial index set $J_1 := \{1\}$ and the initial prox-center to $\hat{x}^1 := x^1$. Set $\hat{f}_1 = f_1$, $s_1^1 = g^1$ and select $t_1 > 0$. Initialize $c_1^1 = 0$ and $M_1(\hat{x}^1 + d) = \hat{f}_1 + \langle s_1^1, d \rangle$.
For $k = 1, 2, 3, \ldots$

1. Calculate

$$d^k = \underset{d \in \mathbb{R}^n}{\arg\min} \left\{ M_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d \right\rangle \right\}.$$

2. Set

$$G^k = \sum_{j \in J_k} \alpha_j^k s_j^k,$$
$$C_k = \sum_{j \in J_k} \alpha_j^k c_j^k \text{ and}$$
$$\delta_k = C_k + \left\langle d^k, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle.$$

   If $\delta_k \leq \texttt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.

4. Compute f^{k+1}, g^{k+1}.
   If $f^{k+1} \leq \hat{f}^k - m\delta_k \quad \rightarrow$ serious step:
   Set $\hat{x}^{k+1} = x^{k+1}, \hat{f}^{k+1} = f^{k+1}$ and calculate a symmetric matrix Q_{k+1} with $-q\mathbb{I} \preccurlyeq Q_{k+1} \preccurlyeq q\mathbb{I}$.
   Adjust t_{k+1} such that $Q_{k+1} + \frac{1}{t_{k+1}}\mathbb{I} \succ 0$ and $t_{k+1} > t_{min}$.
   Otherwise $\quad \rightarrow$ nullstep:
   Set $\hat{x}^{k+1} = \hat{x}^k, \hat{f}^{k+1} = \hat{f}^k$ and choose $t_{min} < t_{k+1} \leq t_k$.

5. Select the new bundle index set J_{k+1}. Calculate

$$\eta_{k+1} = \max\left\{ \max_{j \in J_{k+1}, x^j \neq \hat{x}^{k+1}} \frac{-2e_j^{k+1}}{|x^j - \hat{x}^{k+1}|^2}, 0 \right\} + \gamma$$

   and c_j^{k+1} by (5.5) for all $j \in J^{k+1}$. Update the model M_{k+1} such that conditions (5.14) and (5.15) are fulfilled.

6.3. Convergence Analysis

In this chapter the convergence properties of the new method are analyzed. We do this the same way it is done by Hare et al. in [16].

In the paper all convergence properties are first stated in [16, Lemma 5]. It is then shown that all sequences generated by the method meet the requirements of this lemma which is stated as Lemma 5.2 in this thesis.

As neither the stabilization nor the descent test are involved in the proof of Lemma 5.2 it is the same as in [16].

We prove now that also the variable metric version of the algorithm fulfills all requirements

of Lemma 5.2. The proof is divided into two parts. The first case covers the case of infinitely many serious steps, the second one considers infinitely many null steps after a finite number of serious steps.

For both proofs the equivalence of norms is used between the Euclidean norm and the norm $\|\cdot\|_{Q_k+\frac{1}{t_k}\mathbb{I}}$. We show here shortly, that the matrix $Q_k+\frac{1}{t_k}\mathbb{I}$ can be used to define a scalar product which induces the norm $\|\cdot\|_{Q_k+\frac{1}{t_k}\mathbb{I}}$.

In order to do this, it is first shown, that the matrix $Q_k+\frac{1}{t_k}\mathbb{I}$ is bounded.

Proposition 6.1 The matrix $Q_k+\frac{1}{t_k}\mathbb{I}$ is bounded in the sense that $\left\|Q_k+\frac{1}{t_k}\mathbb{I}\right\|_2<\infty$ for all k and the spectral norm $\|\cdot\|_2$ [21, Example 5.6.6].

This means also that the following relation holds for all vectors $x\in\mathbb{R}^n$:

$$\left\|\left(Q_k+\frac{1}{t_k}\mathbb{I}\right)x\right\|\leq\left|q+\frac{1}{t_{min}}\right|\|x\|,\quad\forall k, \tag{6.6}$$

where $q>0$ is the bound on the eigenvalues of Q_k and t_{min} the lower bound for the step size.
Inequality (6.6) also holds true in the limit $k\to\infty$.

Proof:

Algorithm 6.1 ensures that $-q\mathbb{I}\preccurlyeq Q_k\preccurlyeq q\mathbb{I}$ for some preset constant $q>0$. This means, that the matrices $Q_k+q\mathbb{I}$ and $q\mathbb{I}-Q_k$ are positive semidefinite yielding that all their eigenvalues are nonnegative.

In section A.1.1 in the appendix it is shown that the eigenvalues of a matrix of the form $A+b\mathbb{I}$ for a symmetric matrix $A\in\mathbb{R}^{n\times n}, b\in\mathbb{R}$ are $\tilde{\lambda}_i:=\lambda_i^A+b$, with λ_i^A, $i=1,...,n$ being the eigenvalues of the matrix A. This means that the following holds for all eigenvalues λ_i^k, $i=1,...,n$, of Q_k, which is symmetric, for all k:

$$\lambda_i^k+q\geq 0 \text{ and } q-\lambda_i^k\geq 0\quad\Rightarrow\quad|\lambda_i^k|\leq q,\quad i=1,...,n,\quad\forall k.$$

This means that also $\lim_{k\to\infty}|\lambda_i^k|\leq q$.

The step size t_k is bounded below by t_{min} for all k and hence also for $k\to\infty$. This yields for the spectral norm that

$$\|Q_k+\frac{1}{t_k}\mathbb{I}\|_2=|\lambda_{max}^k+\frac{1}{t_k}|\leq|q+\frac{1}{t_{min}}|<\infty,\quad\forall k. \tag{6.7}$$

Here λ_{max}^k denotes the eigenvalue of Q_k that fulfills $\lambda_{max}^k = \arg\max_{i\in\{1,\dots,n\}} |\lambda_i^k + \frac{1}{t_k}|$.
The spectral norm is induced by the Euclidean norm thus it is compatible with it. Therefore with relation (6.7) it holds

$$\left\| \left(Q_k + \frac{1}{t_k}\mathbb{I} \right) x \right\| \leq \left\| Q_k + \frac{1}{t_k}\mathbb{I} \right\|_2 \|x\| \leq \left| q + \frac{1}{t_{min}} \right| \|x\|, \quad \forall k.$$

The estimates above were also shown to hold in the limit $k \to \infty$ so the inequalities hold also true in that case. □

Proposition 6.2 The norm $\|\cdot\|_{Q_k+\frac{1}{t_k}\mathbb{I}}$ induced by the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ is well-defined for all k.

The proof to this proposition is found in section A.1.2 of the appendix.

The first part of the convergence proof is now stated.

Theorem 6.3 (c.f. [16, Theorem 6, p. 14]) *Let algorithm 6.1 generate an infinite number of serious steps. Then $\delta_k \to 0$ as $K \ni k \to \infty$ for the sequence of serious steps $K \subset \{1, 2, ...\}$. Let the sequence $\{\eta_k\}$ be bounded. As $K \ni k \to \infty$ we have $C_k \to 0$. For every accumulation point $\bar{x}$ of $\{\hat{x}^k\}$ there exists a subsequence $\tilde{K}$ and a vector $\bar{S}$ such that $S^{\tilde{k}} \to \bar{S}$ and $S^{\tilde{k}} + \nu^{\tilde{k}} \to 0$ for $\tilde{K} \ni \tilde{k} \to \infty$. In particular if the cardinality of $\{j \in J^k \mid \alpha_j^k > 0\}$ is uniformly bounded in k then the conclusions of Lemma 5.2 hold.*

The proof is very similar to the one stated in [16] but minor changes have to be made due to the different formulation of the nominal decrease δ_k.

Proof: Let $K \subset \{1, 2, ...\}$ denote the subsequence of serious steps. (For the sake of readability the same index k that is used in the algorithm for all steps is used here only for the serious steps.) At each of those steps we have

$$\hat{f}_{k+1} \leq \hat{f}_k - m\delta_k, \quad k \in K \tag{6.8}$$

where $m,\ \delta_k > 0$. As $\hat{f}_k$ is only updated in serious steps it follows that the sequence $\{\hat{f}_k\}$ is nonincreasing. Since the sequence $\{\hat{x}^k\}$ lies in the compact set X and f is continuous the sequence $\{f(\hat{x}^k)\}$ is bounded. With $|\sigma_k| < \bar{\sigma}$ also the sequence $\{f(\hat{x}^k) + \sigma_k\} = \{\hat{f}_k\}$ is bounded. Considering also the fact that $\{\hat{f}_k\}$ is nonincreasing one can conclude that it converges.
From (6.8) follows that

$$0 \leq m \sum_{k=1}^{l} \delta_k \leq \sum_{k=1}^{l} \left(\hat{f}_k - \hat{f}_{k+1} \right),$$

so letting $l \to \infty$,

$$0 \leq m \sum_{k=1}^{\infty} \delta_k \leq \hat{f}_1 - \underbrace{\lim_{k \to \infty} \hat{f}_k}_{\neq \pm\infty}.$$

This yields

$$\sum_{k=1}^{\infty} \delta_k = \sum_{k=1}^{\infty} \left(C^k + \left\langle d^k, \left(Q_k + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle \right) < \infty.$$

Hence, $\delta_k \to 0$ as $k \to \infty$. All quantities above are nonnegative due to positive definiteness of $Q_k + \frac{1}{t_k}\mathbb{I}$ and $C_k \geq 0$ so it also holds that

$$C_k \to 0 \quad \text{and} \quad \left\langle d^k, \left(Q_k + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle \to 0.$$

Finally we need to show that for any accumulation point $\bar{x}$ of the sequence $\{\hat{x}^{\tilde{k}}\}$ holds $S^{\tilde{k}} \to \bar{S}$ and $S^{\tilde{k}} + \nu^{\tilde{k}} \to 0$ for $\tilde{K} \ni \tilde{k} \to \infty$ and the suitable subsequence $\tilde{K} \subset K$. Let $K' \subset K$ denote the subset such that the sequence $\{\hat{x}^{k'}\}$ converges to its accumulation point $\bar{x}$ for $k' \in K'$. As we only consider the subsequence of serious steps it follows from $\{\hat{x}^{k'}\}_{k' \in K'} \to \bar{x}$ that $d^{k'} = \hat{x}^{k'+1} - \hat{x}^{k'} \to 0$ for $K' \ni k' \to \infty$. The step size t_k is bounded below by $t_{min} > 0$ and because the eigenvalues of Q_k are bounded the expression

$$S^{k'} + \nu^{k'} = \left(Q_{k'} + \frac{1}{t_{k'}} \mathbb{I} \right) d^{k'} \to 0 \quad \text{for} \quad K' \ni k' \to \infty$$

because

$$\|S^{k'} + \nu^{k'}\| = \left\| \left(Q_{k'} + \frac{1}{t_{k'}} \mathbb{I} \right) d^{k'} \right\| \leq \left| q + \frac{1}{t_{min}} \right| \underbrace{\|d^{k'}\|}_{\to 0}.$$

Here the last inequality follows from (6.6). The implication $S^{\tilde{k}} \to \bar{S}$ for $\tilde{k} \in \tilde{K}$ follows from local Lipschitz continuity of f. By Rademacher's theorem this property yields that on any open set Ω the function f is differentiable except on a set Ω_{nd} of zero Lebesgue measure. As f is also Lipschitz continuous on any closed set containing such an open set Ω, the gradient of f on Ω is bounded. Let $X \subset \Omega$. By theorem 2.5.1 in [4, p. 63] the subdifferential of f at any point $x^k \in \Omega$ is the convex hull of the limits of gradients ∇f

of f on the set $\Omega \setminus \Omega_{\text{nd}}$

$$\partial f(x^k) = \text{conv}\{\lim \nabla f(y) \mid y \to x^k, y \notin \Omega_{\text{nd}}\}.$$

This means that also all subgradients on Ω are bounded. As the subgradient error is assumed to be bounded by $\bar{\theta}$ the set of approximate subgradients $\{g^j, j \in J^k\}$ contained in the bundle is bounded as well. From this follows that also the augmented subgradients $s_j^k = g^j + \eta_k(x^j - \hat{x}^k)$ are bounded because η_k is bounded by assumption and $x^j, \hat{x}^k \in X$. Defining $s := \max_{k,j} \|s_j^k\|$ this yields that

$$\|S^k\| = \left\| \sum_{j \in J^k} \alpha_j^k s_j^k \right\| \leq \sum_{j \in J^k} \|\alpha_j^k\| \underbrace{\|s_j^k\|}_{\leq s \in \mathbb{R}} \leq s \underbrace{\sum_{j \in J^k} \alpha_j^k}_{=1} < \infty \quad \forall k.$$

It follows that the sequence S^k is bounded. By the Bolzano-Weierstrass theorem [27, p. 51] every bounded sequence has a convergent subsequence. Let $\tilde{K} \subset \hat{K}'$ denote the index set of this converging subsequence and $\bar{S}$ the corresponding accumulation point. Then finally $S^{\tilde{k}} \to \bar{S}$ for $\tilde{K} \ni \tilde{k} \to \infty$.

□

For the case that only finitely many serious steps are executed we need the following result:

Whenever x^{k+1} is as declared a null step, a simple calculation shows that the relation

$$-c_{k+1}^{k+1} + \left\langle s_{k+1}^{k+1}, x^{k+1} - \hat{x}^k \right\rangle = f_{k+1} - \hat{f}_k + \underbrace{\frac{\eta_{k+1}}{2} \|x^{k+1} - \hat{x}^k\|^2}_{\geq 0} > -m\delta_k \tag{6.9}$$

holds. The exact derivation of (6.9) is also given in [16, p. 16].

Another relation that is used a few times throughout the proof is the estimate

$$\left\langle \nu^k, d^k \right\rangle \geq 0. \tag{6.10}$$

It follows from the subgradient inequality for the convex function $\mathtt{i}_X$ at the point x^{k+1}. As $\nu^k \in \partial \mathtt{i}_X(x^{k+1})$ it holds $\mathtt{i}_X(y) - \mathtt{i}_X(x^{k+1}) \geq \left\langle \nu^k, y - x^{k+1} \right\rangle$ for all $y \in X$ and as $d^k = x^{k+1} - \hat{x}^k$ and $x^{k+1}, \hat{x}^k \in X$ it follows

$$0 = \underbrace{\mathtt{i}_X(\hat{x}^k)}_{=0} - \underbrace{\mathtt{i}_X(x^{k+1})}_{=0} \geq \left\langle \nu^k, \hat{x}^k - x^{k+1} \right\rangle = - \left\langle \nu^k, d^k \right\rangle$$

yielding inequality (6.10) above.

Theorem 6.4 (c.f. [16, Theorem 7, p. 16]) *Let a finite number of serious iterates be followed by infinite null steps. Let the sequence $\{\eta_k\}$ be bounded. Then $\{x^k\} \to \hat{x}$, $\delta_k \to 0$, $C_k \to 0$, $S^k + \nu^k \to 0$ and there exist $K \subset \{1, 2, ...\}$ and $\bar{S}$ such that $S^k \to \bar{S}$ as $K \ni k \to \infty$.*
In particular if the cardinality of $\{j \in J^k \mid \alpha_j^k > 0\}$ is uniformly bounded in k then the conclusions of Lemma 5.2 hold for $\bar{x} = \hat{x}$.

Proof: Let k be large enough such that $k \geq \bar{k}$, where $\bar{k}$ is the iterate of the last serious step. Let $\hat{x} := \hat{x}^{\bar{k}}$ and $\hat{f} := \hat{f}_{\bar{k}}$ be fixed. The matrix Q_k is also fixed and denoted as $Q := Q_{\bar{k}}$. Define the optimal value of the k'th subproblem (6.3) for $k > \bar{k}$ by

$$\Psi_k := M_k(x^{k+1}) + \frac{1}{2} \left\langle d^k, \left(Q + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle. \tag{6.11}$$

It is first shown that the sequence $\{\Psi_k\}$ is bounded above. From definition (5.9) and relation (5.13) follows

$$A_k(\hat{x}) = M_k(x^{k+1}) - \left\langle S^k, d^k \right\rangle \leq M_k(\hat{x}).$$

Using (6.4) for the third equality and (6.10) in the first inequality one obtains

$$\begin{aligned}
\Psi_k + \frac{1}{2} \left\langle d^k, \left(Q + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle &= A_k(\hat{x}) + \left\langle S^k, d^k \right\rangle + \left\langle d^k, \left(Q + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle \\
&= A_k(\hat{x}) + \left\langle S^k + \langle Q + \frac{1}{t_k} \mathbb{I}, d^k \rangle, d^k \right\rangle \\
&= A_k(\hat{x}) - \left\langle \nu^k, d^k \right\rangle \\
&\leq A_k(\hat{x}) \\
&\leq M_k(\hat{x}) \\
&= \hat{f}.
\end{aligned}$$

By boundedness of d^k and boundedness and positive definiteness $Q + \frac{1}{t_k} \mathbb{I}$ this yields that

$\Psi_k \leq \Psi_k + \frac{1}{2}\|d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}} \leq \hat{f}$, so the sequence $\{\Psi_k\}$ is bounded above. In the next step it is shown that $\{\Psi_k\}$ is increasing. By noting that $x^{k+2} = \hat{x} + d^{k+1}$, as the proximal center does not change in the null step case, we obtain

$$
\begin{aligned}
\Psi_{k+1} &= M_{k+1}(x^{k+2}) + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_{k+1}})\mathbb{I}\right) d^{k+1}\right\rangle \\
&\geq A_k(\hat{x} + d^{k+1}) + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&= M_k(x^{k+1}) + \left\langle S^k, d^{k+1} - d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&= M_k(x^{k+1}) + \left\langle -\left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k - \nu^k, d^{k+1} - d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&= \Psi_k - \frac{1}{2}\left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&\quad - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^{k+1} - d^k\right)\right\rangle - \left\langle \nu^k, d^{k+1} - d^k\right\rangle \\
&= \Psi_k + \frac{1}{2}\left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&\quad - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle - \left\langle \nu^k, x^{k+2} - x^{k+1}\right\rangle \\
&\geq \Psi_k + \frac{1}{2}\left\langle \left(d^{k+1} - d^k\right), \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^{k+1} - d^k\right)\right\rangle \\
&= \Psi_k + \frac{1}{2}\underbrace{\|d^{k+1} - d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}}}_{\geq 0}.
\end{aligned}
\tag{6.12}
$$

Here the first inequality comes from (5.13) and the fact that $t_{k+1} \leq t_k$ for null steps. The second equality follows from (5.9), the fourth equality by (6.4) and (6.11) and the last inequality holds by the subgradient inequality for $\nu^k \in \mathtt{i}_X(x^{k+1})$ and the fact that $x^{k+1}, x^{k+2} \in X$.

Looking again at (6.12) and taking into account that $1/t_k \geq 1/t_{\bar{k}}$ in the null step case we have

$$
\begin{aligned}
\Psi_{k+1} - \Psi_k &\geq \frac{1}{2}\|d^{k+1} - d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}} \\
&\geq \frac{1}{2}\|d^{k+1} - d^k\|^2_{Q+\frac{1}{t_{\bar{k}}}\mathbb{I}} \geq 0.
\end{aligned}
$$

This means that the sequence $\{\Psi_k\}$ is increasing and bounded from above. Thus the sequence is convergent.

This yields

$$|\Psi_{k+1} - \Psi_k| \to 0 \quad \Rightarrow \quad \|d^{k+1} - d^k\| \to 0 \text{ for } k \to \infty \tag{6.13}$$

due to the equivalence of norms.

By the last line in (6.5) and the fact that $\hat{f} = M_k(\hat{x})$ for all $k > \bar{k}$ we have

$$\begin{aligned}
\hat{f} &= M_k(\hat{x}) + \delta_k - C_k - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \\
&= M_k(\hat{x}) - \hat{f} + M_k(x^{k+1}) + \delta_k - \left\langle S^k, d^k \right\rangle - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \\
&= \delta_k + M_k(\hat{x} + d^k) + \left\langle \nu^k, d^k \right\rangle \\
&\geq \delta_k + M_k(\hat{x} + d^k),
\end{aligned}$$

where the second equality is by (5.12), the third holds because of relation (6.4) and the last inequality is given by (6.10). Therefore

$$\delta^{k+1} \leq \hat{f} - M_{k+1}(\hat{x} + d^{k+1}). \tag{6.14}$$

By assumption (5.14) on the model, written for $d = d^{k+1}$,

$$-\hat{f}_{k+1} + c_{k+1}^{k+1} - \left\langle s_{k+1}^{k+1}, d^{k+1} \right\rangle \geq -M_{k+1}(\hat{x} + d^{k+1}).$$

In the null step case it holds $\hat{f}_{k+1} = \hat{f}$ so combining condition (6.9) and the inequality above, one obtains that

$$m\delta_k + \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \geq \hat{f} - M_{k+1}(\hat{x} + d^{k+1}).$$

In combination with (6.14) this yields

$$0 \leq \delta_{k+1} \leq m\delta_k + \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \leq m\delta_k + \left| \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \right|. \tag{6.15}$$

For the next step Lemma 3 and the corollary below it from [49, p. 45] are used. They

state that for

$$u_{k+1} \leq q u_k + a_k, \quad q < 1, \quad a_k \geq 0, \quad a_k \to 0 \text{ and } u_k \geq 0$$

it holds $u_k \to 0$.

Taking the first and the last part of inequality (6.15) we can identify $u_k = \delta_k \geq 0$, $q = m \in (0,1)$ and $a_k = \left|\left\langle s_{k+1}^{k+1}, d^k - d^{k+1}\right\rangle\right| \geq 0$. To show that $a_k \to 0$ recall (6.13) and that the augmented subgradient s_{k+1}^{k+1} is bounded due to local Lipschitz continuity of f and boundedness of $\{\eta_k\}$ by the same argumentation as in the case of infinitely many serious steps.

The lemma then gives that

$$\lim_{k\to\infty} \delta_k = \lim_{k\to\infty} C_k + \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle = 0.$$

By the fact that $C_k \geq 0$ for all k and positive definiteness of $Q + \frac{1}{t_k}\mathbb{I}$ it follows that all summands above are nonnegative and hence $C_k \to 0$ as $k \to \infty$. As the matrix $Q + \frac{1}{t_k}\mathbb{I}$ is bounded due to $t_k > t_{min} > 0$ and the bounded eigenvalues of Q and because in null steps $t_k \leq t_{\bar{k}}$,

we have

$$\|d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}} \geq \|d^k\|^2_{Q+\frac{1}{t_{\bar{k}}}\mathbb{I}} \geq c\|d^k\|^2 \to 0.$$

for a constant $c \in \mathbb{R}$ by the equivalence of norms.

This means that $d^k \to 0$ for $k \to \infty$ and therefore $\lim_{k\to\infty} x^k = \hat{x}$. It also follows that $\|S^k + \nu^k\| \to 0$ as $k \to \infty$ because of

$$\|S^k + \nu^k\| = \left\|\left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k\right\| \leq \left|q + \frac{1}{t_{min}}\right| \underbrace{\|d^k\|}_{\to 0} \to 0.$$

By the same arguments as in the proof of Theorem 6.3 the local Lipschitz property of the objective function f and boundedness of the subgradient errors θ_k yield boundedness of the sequence S^k. Passing to some subsequence $K \subset \{1, 2, ...\}$ if necessary we can therefore conclude that the sequence $\{S^k\}_{k\in K}$ converges to some $\bar{S}$ and as $\hat{x}^k = \bar{x}$ for all $k \geq \bar{k}$ all requirements of Lemma 5.2 are fulfilled.

□

Remark: In case the matrix Q_k is also updated in null steps the proof still holds as long as the assumptions on boundedness of Q_k and especially positive definiteness of $Q_k + \frac{1}{t_k}\mathbb{I}$ even in the limit $k = \infty$ are still valid.

Remark: All results deduced in section 5.2 are still valid for this algorithm as they do not depend on the kind of stabilization used.

6.4. Updating the Metric

In [47] and [45] it is not specified how the matrices Q_k are chosen. For convergence it is necessary that Q_k is symmetric and its eigenvalues are bounded. Here we present some possibilities to update the metric matrix Q_k such that it fulfills both conditions.

Most of the presented updates are based on the BFGS-update formula (named after Broyden, Goldfarb, Fletcher and Shanno)

$$\tilde{Q}_{k+1} = Q_k + \frac{y^k {y^k}^\top}{\langle y^k, d^k \rangle} - \frac{Q_k d^k (Q_k d^k)^\top}{\langle d^k, Q_k d^k \rangle}. \tag{6.16}$$

Usually y^k is defined as the difference of the last two gradients of f. To adapt the formula to the nondifferentiable case the difference $y^k := g^{k+1} - g^k$ of two (approximate) subgradients of f is taken instead as proposed in [13]. The starting matrix is $Q_1 = \mathbb{I}$.

By definition the BFGS update is symmetric. To assure boundedness of the matrix Q_{k+1} the updates can be manipulated in the following ways:

6.4.1. Scaling of the Whole Matrix

A simple way to keep the absolute value all of eigenvalues of the constructed matrix Q_k below some threshold $0 < q < \infty$ is to scale the whole matrix down as soon as the absolute value of one eigenvalue is larger than this number. To do this define $\lambda_{max} := \max\{|\lambda_i^k| \mid \lambda_i^k \text{ is eigenvalue of } \tilde{Q}_k\}$. If $\lambda_{max}^k > q$, set $Q_k = \frac{q}{\lambda_{max}^k}\tilde{Q}_k$, where $\tilde{Q}_k$ is the matrix coming from the BFGS update. This way the absolute value of all eigenvalues is always smaller or equal to q. An advantage of this method is besides its simplicity that by scaling the whole matrix the ratio of the eigenvalues of Q_k is preserved. Scaling of Q_k corresponds to shrinking the whole quadratic function and in this way also the 'ratio of curvature' at different points of the graph stays the same.

6.4.2. Adaptive Scaling of Single Eigenvalues

The second method is motivated by some properties of lower-$\mathcal{C}^2$ functions. This function class is very suitable to be used with the presented bundle algorithm and there exist some practical applications that entail such functions (c.f. [15] and [16]).

Lower-$\mathcal{C}^2$ functions can locally be written as the maximum over finitely many $\mathcal{C}^2$ functions.

For the following motivation let us consider a lower-$\mathcal{C}^2$ function $f : \mathbb{R} \to \mathbb{R}$ on an open set $\Omega \subset \mathbb{R}$ that can be written as

$$f(x) = \max_{t \in T} f_t(x) \tag{6.17}$$

for a finite index set T fulfilling the conditions of Definition 2.5.

Consider from now on the function f on the set Ω. At points where the maximum in (6.17) is only attained by a single function f_t the function f is twice continuously differentiable and hence provides curvature information. At points where the are more than one function attaining the same value this does not have to be the case. At those points f can be nondifferentiable.

Consider a nondifferentiable point and denote it as $x_{\text{kink}} \in \Omega$. As f is locally Lipschitz continuous Rademacher's theorem yields that the points in Ω where the function f is nondifferentiable are a set of zero Lebesgue measure. This means that there exists an open interval with length $2r$, $r > 0$, around the point x_{kink} such that $(x_{\text{kink}} \pm r) \in \Omega$ and the function f is differentiable on the two open intervals $(x_{\text{kink}} - r, x_{\text{kink}})$ and $(x_{\text{kink}}, x_{\text{kink}} + r)$. Assume, that f is is *directionally differentiable* in x_{kink} and that all the directional derivatives in that point are finite. The directional derivative with respect to the direction $d \in \mathbb{R}^n$ is defined as [50, p. 213]

$$f'(x, d) = \lim_{h \searrow 0} \frac{f(x + hd) - f(x)}{h}.$$

The function f has for example finite directional derivatives in x_{kink}, if it is convex on Ω [56, p. 144].

In the one dimensional case this means that we can denote the *right derivative* $f(x_{\text{kink}}, 1) := a$ and the *left derivative* $-f(x_{\text{kink}}, -1) := b$. As f was assumed to be nondifferentiable in x_{kink} it holds that $a \neq b$ [50, p. 213].

To get a hint on what we can expect from the 'curvature' at that kink the following

quotient is examined:

$$\lim_{h \searrow 0} \frac{\overbrace{f'(x_{\text{kink}} - h, 1)}^{\to a} + \overbrace{f'(x_{\text{kink}} + h, -1)}^{\to b}}{h} = \pm\infty.$$

This holds by continuity of the of the derivatives on both sides of the kink and the directional derivatives being the respective limits at the kink. The sign depends on the signs of a and b.

In more dimensions this is the same for the components of the metric matrix Q_k corresponding to the direction where the kink occurs. This supports also to the intuitive thought that at a kink the slope changes 'infinitely fast'. Numerically the BFGS-update (6.16) can result in very large values for the entries of Q_k corresponding to points near the kink.

On the other hand due to the local Lipschitz property the slope of the objective function is always finite on closed sets. This means that there exists an interval $(x_{\text{kink}} - r', x_{\text{kink}} + r')$, $r' \in \mathbb{R}$, where the function f behaves similar to the scaled modulus $a|\cdot|$, $a \in \mathbb{R}$, in the direction perpendicular to the kink. Therefore in this neighborhood almost no curvature is present.

Summarized this means that on the one hand, the matrix Q_k should be close to zero in the components representing the directions perpendicular to the kink as soon as the iterates approach x_{kink}. But contrary to that the method that constructs Q_k can give very high values for those components.

Remark: The above considerations are only a motivation for the following practical matrix update. A more rigorous theoretical background for the update is still open.

Let again $\tilde{Q}_k$ denote the matrix coming directly form the BFGS update. The idea is now to scale only those eigenvalues of $\tilde{Q}_k$ that are especially large. To do this calculate all eigenvalues λ_i^k of $\tilde{Q}_k$. As the metric matrix $\tilde{Q}_k$ is a symmetric real matrix it is always orthogonally diagonalizable [35, Corollary 18.18 p. 282]. An eigenvalue decomposition $\tilde{Q}_k = U \cdot \tilde{D} \cdot U^\top$ is available. The diagonal matrix $\tilde{D}$ has the eigenvalues of $\tilde{Q}_k$ on its diagonal and the transformation matrix U contains the corresponding eigenvectors.

Then all eigenvalues of $\tilde{Q}_k$ that are larger than q are scaled and replaced in the matrix $\tilde{D}$. The bounded version of $\tilde{Q}_k$ is obtained by transforming the bounded matrix D back into the full matrix with help of the transformation matrix U. Let D denote the matrix with the scaled eigenvalues on the diagonal. The matrix U is not changed. This means

that $Q_k := A \cdot D \cdot A^\top$ has the same eigenvectors as the original update $\tilde{Q}_k$ but bounded eigenvalues.

The above two methods were tested in practice and the results of the algorithm are shown in section 6.5. We also compare them to a hybrid method where the first approach is used for the updates and the matrix is additionally scaled by the stepsize such that the final metric matrix is $Q_k = \frac{1}{k}\bar{Q}_k$ with $\bar{Q}_k$ being the scaled BFGS update suggested in section 6.4.1. This way the method starts out as the variable metric method but becomes more equal to the proximal bundle method 5.1 as the algorithm continues.

Remark: There appear many parameters to control the scaling of the metric update. Although these parameters were not especially tuned in this thesis it was observed that they have a considerable impact on the convergence speed of the method also depending on the objective function used. This has to be kept in mind when implementing the method in practice.

6.4.3. Other Updating Possibilities

There are certainly many other possibilities to update the metric Q_k. A third variation based on BFGS-updates is the limited memory update suggested in [43]. If the update is skipped whenever $\frac{\|d^k\|}{\|y^k\|} > \tilde{q}, \tilde{q} > 0$, the matrix Q_k stays bounded. (Remark that in general $\tilde{q} \neq q$, so it is not directly the lower bound for the step size t_{min}.) This strategy is also supported by the fact that if $\frac{\|d^k\|}{\|y^k\|} > \tilde{q}$ the change in the subgradient relative to the step size is rather small indicating that the current iterate lies within a region with only small changes in curvature. In such regions the update can be skipped. It is also possible to alter the updates presented above by a special choice of the subgradients. For example trying to compute the directional derivative or using more information by considering more subgradients of the bundle.

Another updating method is using the symmetric-rank-1 (SR1) update

$$\tilde{Q}_k = Q_{k-1} + \frac{(y^k - Q_k d^k)(y^k - Q_k d^k)^\top}{\langle y^k - Q_k d^k, d^k \rangle}.$$

Boundedness can be assured in the same way as for the BFGS update.

Finally the strategies to measure the need of scaling of the matrices to ensure boundedness are diverse. Here it could be interesting to consider the change in the matrix $\tilde{Q}_k$ relative to Q_{k-1} instead of using the absolute values of the eigenvalues as a threshold. This is however hardly possible if the eigenvalues themselves are taken as a measure as they lack

an intrinsic order. This makes it hard to find the corresponding eigenvalues λ_i^{k-1} and λ_i^k in consecutive updates in order to compute the change between them.

In higher dimensions updating the matrix Q_k can be costly. This is one of the reasons why it is not updated in null steps. Also in null steps the proximal center stays the same, so it can be assumed that not much curvature information can be gained when updating during such steps. Still updating in null steps has the advantage of making use of the additional subgradient information provided in those steps. In [13], where the metric matrix is updated also in null steps, a BFGS update is used in serious steps and the less costly SR1 update in null steps.

Remark: Bounding the eigenvalues of Q_k by $q \in \mathbb{R}$ also assures that t_k can be bounded from below without impairing positive definiteness of the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$. This can be done by setting $t_{min} = \frac{1}{q} - \varepsilon$ for a small positive constant ε as it is done in algorithm 6.

As a last remark on this topic we want to say that although in this thesis the adaption of update strategies originally developed to be used with gradients seems to work in the presented setting also with subgradients this does not always have to be the case. Although it is argued for example in [32] that locally Lipschitz functions are differentiable almost everywhere and with an adequate linesearch it is improbable to arrive at an iterate that is a nondifferentiable point of the objective function this can still happen. Especially if such a linesearch is not used like in the algorithm presented here. So despite the promising practical behavior this area is still open to research.

6.5. Numerical Tests

To compare the proximal bundle algorithm 5.1 with its variable metric variant algorithm 6.1 both are tested on some academic test functions and on a set of lower-$\mathcal{C}^2$ functions in different dimensions. The tests are done with the following parameters given in [16]: $m = 0.05$, $\gamma = 2$ and $t_0 = 0.1$. The chosen stopping tolerance is $\texttt{tol} = 10^{-6}$. If the algorithms do not meet the stopping condition after $250n$ steps for $x \in \mathbb{R}^n$, they are terminated. Contrary to [16] the stopping test is taken as given in the algorithm and the tolerance not multiplied by $1 + \hat{f}_k$. The proximity control parameters κ_- and κ_+ from (3.23) and (3.22) respectively are chosen as $\kappa_- = 0.8$ and $\kappa_+ \in \{1.2, 2\}$. When the bundle is updated at the end of each iteration additionally to the newly computed iterate the current prox-center and all elements that have corresponding Lagrange multipliers $\alpha_j^k > 10^{-15}$ are kept in the bundle.

In the metric matrix updates the threshold q is chosen 10^8. For the adaptive variant of the update eigenvalues that are larger than q are set to $q/10$.

The algorithms are abbreviated in the legends of the plots as 'Bundle Nonconv Inex' for the proximal bundle algorithm 5.1 and 'Variable Metric BFGS' and 'Variable Metric BFGS Adaptive' for the variable metric bundle algorithm 6.1 using the scaled update and the adaptive eigenvalue scaling respectively.

To test the performance for inexact function and subgradient values different types of noise are introduced. This is done by adding randomly generated elements with norm less or equal to σ_k and θ_k to the exact values $f(x^{k+1})$ and $g(x^{k+1})$ respectively.

Five different forms of noise are tested:

- N_0: No noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = \theta_k = 0$ for all k,
- $\mathrm{N}_\mathrm{c}^{\mathrm{f,g}}$: Constant noise, $\bar{\sigma} = \sigma_k = 0.01$ and $\bar{\theta} = \theta_k = 0.01$ for all k,
- $\mathrm{N}_\mathrm{v}^{\mathrm{f,g}}$: Vanishing noise, $\bar{\sigma} = 0.01, \sigma_k = \min\{0.01, \|x^k\|/100\}$ and $\bar{\theta} = 0.01, \theta_k = \min\{0.01, \|x^k\|/100\}$ for all k,
- $\mathrm{N}_\mathrm{c}^{\mathrm{g}}$: Constant subgradient noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = \theta_k = 0.01$ for all k and
- $\mathrm{N}_\mathrm{v}^{\mathrm{g}}$: Vanishing subgradient noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = 0.01, \theta_k = \min\{0.01, \|x^k\|/100\}$ for all k.

The exact case is used for comparison. The constant noise forms represent cases where the inexactness is outside of the optimizer's control. The vanishing noise forms represent cases where the noise can be controlled but the mechanism is considered expensive, so it is only used when approaching the minimum. The two forms of subgradient noise represent the case where the subgradient is approximated numerically.

To compare the performance of the different methods the accuracy is measured by

$$\text{accuracy} = |\log_{10}(\hat{f}_{\bar{k}})|.$$

Here $\hat{f}_{\bar{k}}$ is the current $\hat{f}_k$ when the algorithm stops.

All calculations were performed on an Intel i5 2.6 GHz with four kernels.

6.5.1. Academic Test Examples

For the comparison in this section the proximal bundle method and the variable metric method with the two BFGS update rules for Q_k presented in section 6.4 are used.

To explore the benefit of the matrix Q_k the algorithms 5.1 and 6.1 are tested on a smooth and a nonsmooth version of a badly conditioned parabola. The smooth test function is

$$p(x) : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \langle x, Ax \rangle ,$$

where the matrix is chosen as $A = \begin{pmatrix} 1 & 0 \\ 0 & 50 \end{pmatrix}$. The condition number of this matrix is $\kappa_A = \frac{\lambda_{max}}{\lambda_{min}} = 50$, where $\lambda_{min}, \lambda_{max}$ are the smallest and largest eigenvalue of A respectively. From smooth optimization it is known that gradient descent methods have a rather poor convergence rate for such badly conditioned matrices (c.f. chapter 7.4 in [59]). Figure 2 shows the sequences of serious iterates resulting from the two algorithms on the contour lines of the parabola. On the left the complete sequence is depicted. The plot on the right shows a detail of the left figure near the minimum of the objective. As the descent direction taken in algorithm 5.1 is an aggregate subgradient and second order information is only provided by the stabilization term $\frac{1}{t_k}\|d\|^2$ we can see a zig-zagging behavior of the sequence for the parabola in Figure 2. Contrary to that the sequence of serous iterates provided by algorithm 6.1 can take advantage of the second order information provided by Q_k. It approaches the minimum almost in a straight line. The difference in behavior of the two algorithms is especially visible on the detail plot of Figure 2 that shows the situation near the minimum: The proximal bundle algorithm needs a lot of steps circling around the minimum whereas the variable metric algorithm approaches the minimum directly. The resulting advantage of this behavior is the smaller number of steps needed by the variable metric algorithm.

The second test function is a nonsmooth version of the above parabola. The function is given by

$$p_n(x) : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \frac{1}{2} \langle x, Ax \rangle + \frac{1}{2}|x_1| + 25|x_2|.$$

Due to the kink along the x_1-axis the curvature information supplied by Q_k is less reliable than for the smooth parabola. Figure 3 shows the sequences constructed by the two algorithms. Still the sequence provided by the variable metric algorithm does less zig-zagging than the one coming from the proximal bundle algorithm. It is interesting to note, that the sequence provided by the proximal bundle algorithm is the same for both functions. This is not the case for the sequence generated by the metric bundle algorithm because the second order information of the two objective functions is different.

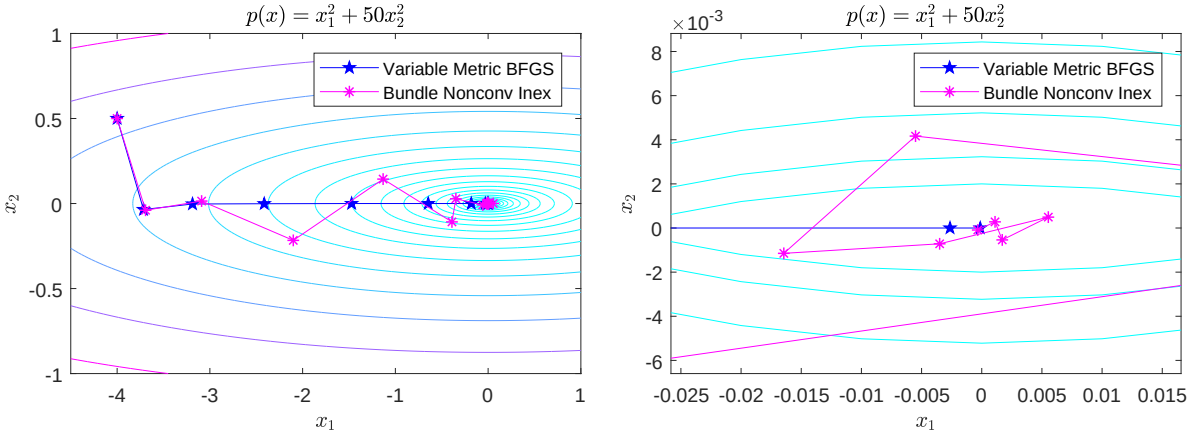

Figure 2: *Sequences of serious steps constructed by the proximal bundle algorithm and the variable metric algorithm respectively on the level lines of parabola p. The right image is a detail of the plot on the left.*
Step size parameter: $\kappa_+ = 2$ *for both algorithms.*

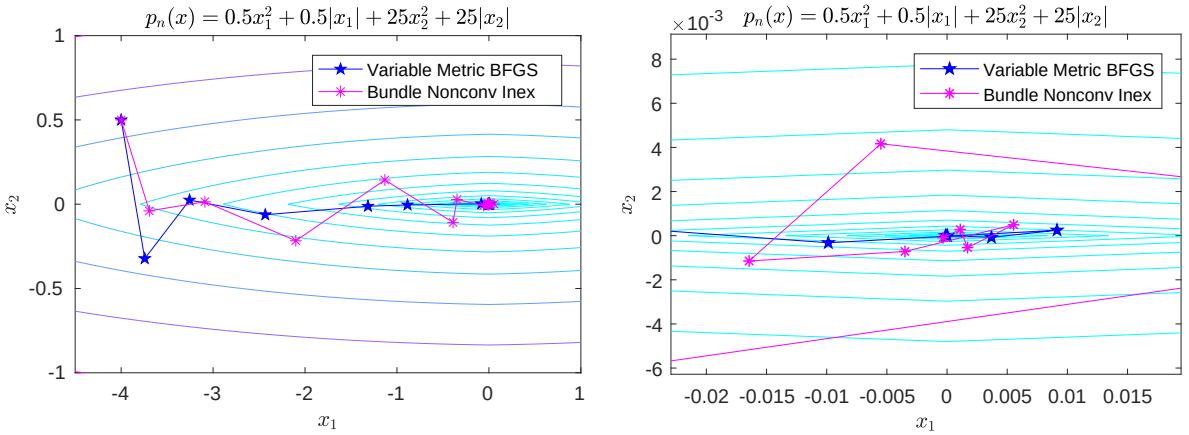

Figure 3: *Sequences of serious steps constructed by the proximal bundle algorithm and the variable metric algorithm respectively on the level lines of the nonsmooth quadratic function* p_n*. The right image is a detail of the plot on the left.*
Step size parameter: $\kappa_+ = 2$ *for both algorithms.*

The bar plots in figures 4 and 5 compare the accuracy of the solution and the number of steps that is needed by the different algorithms for the various noise forms. Here the nonconvex proximal bundle algorithm is compared to both variants of the variable metric method.

To address the random nature of the noise the tests are performed 20 times and the results averaged. The number of steps is rounded to end up with integers.

In the smooth case one can see that the accuracy of the two algorithms is comparable.

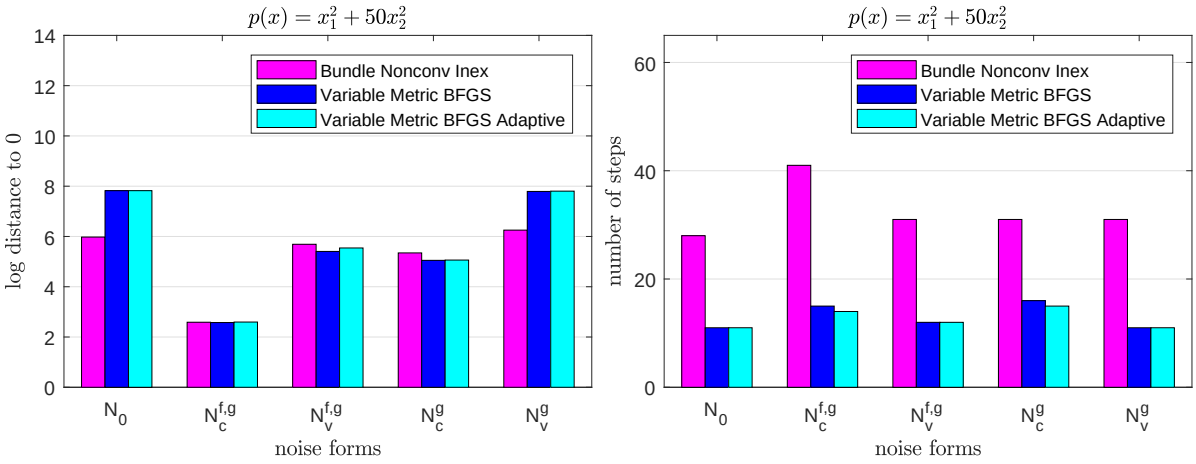

Figure 4: *Left: Accuracy of the solution computed by the different versions of the variable metric bundle algorithm compared to the proximal bundle algorithm for the parabola p under different form of noise.*
Right: Comparison of the number of steps for the three algorithms.
Step size parameters: $\kappa_+ = 1.2$ for the proximal bundle method and $\kappa_+ = 2$ for the variable metric algorithm.

In the case where no noise is present and in the last case, which is the case with the least noise, the variable metric algorithm solves more accurately but for the proximal bundle algorithm the actually computed optimal value is still above the chosen tolerance of 10^{-6}. In the cases of the more involved noise the accuracy is less.

A significant difference can be seen between the needed number of steps of the different algorithms. Here the variable metric versions of the bundle method can take advantage of the curvature information and the fact that for the smooth parabola the BFGS update approximates the Hessian matrix very well. The difference in steps between the two update variants of the variable metric algorithm is neglectable and could also be present due to the random noise. This is what we expect as the scaling mechanism should not be invoked in the smooth case.

In the nonsmooth case (shown in Figure 5) the accuracy of all algorithms is very similar. The difference in the number of steps is now very small as well. Only in the case of constant noise the proximal bundle algorithm performs rather badly. Here the number of steps is extremely large (over 250) in order to gain the same accuracy as the other algorithms. The difference between the two update versions of the variable metric algorithm is still very small. It seems that the different scaling strategies have only a minor influence on the algorithm for this kind of objective function. Other tests showed that for example the choice of the step size updating parameters κ_+, κ_- have a lot more influence

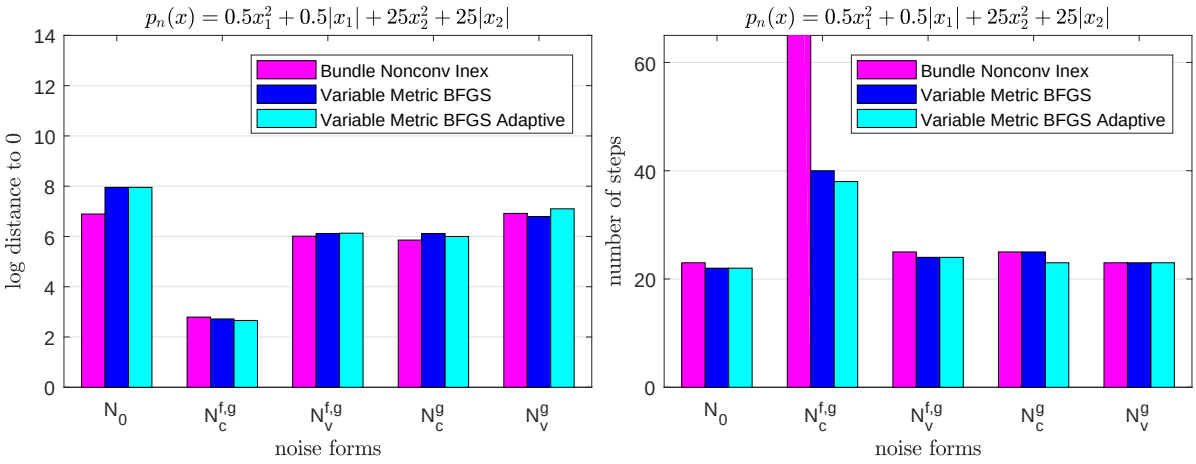

Figure 5: *Left: Accuracy of the solution computed by the different versions of the variable metric bundle algorithm compared to the proximal bundle algorithm for the nonsmooth quadratic function p_n under different form of noise.*
Right: Comparison of the number of steps for the three algorithms. The left bar for the number of steps in the case of constant noise is cropped.
Step size parameters: $\kappa_+ = 1.2$ for the proximal bundle method and $\kappa_+ = 2$ for the variable metric algorithm.

on the algorithm than the tested updating strategies. This can be seen in figures 10 and 11.

6.5.2. Test Examples in Higher Dimensions

For the second test, which involves testing the performance of the different algorithms in different dimensions, the Ferrier polynomials are chosen as objective functions. These nonsmooth and nonconvex functions have already been used in [15] and [16]. The polynomials are constructed in the following way:

For $i = 1, ..., n$ we define

$$h_i : \mathbb{R}^n \to \mathbb{R}, \quad h(x) = (ix_i^2 - 2x_i) + \sum_{j=1}^{n} x_j.$$

These functions are used to define

$$f_1(x) := \sum_{i=1}^{n} |h_i(x)|,$$

$$
\begin{aligned}
f_2(x) &:= \sum_{i=1}^{n} (h_i(x))^2, \\
f_3(x) &:= \max_{i \in \{1,\ldots,n\}} |h_i(x)|, \\
f_4(x) &:= \sum_{i=1}^{n} |h_i(x)| + \frac{1}{2}\|x\|^2 \text{ and} \\
f_5(x) &:= \sum_{i=1}^{n} |h_i(x)| + \frac{1}{2}\|x\|.
\end{aligned}
$$

The graphs of the Ferrier polynomials for $x \in \mathbb{R}^2$ are shown in Figure 6.

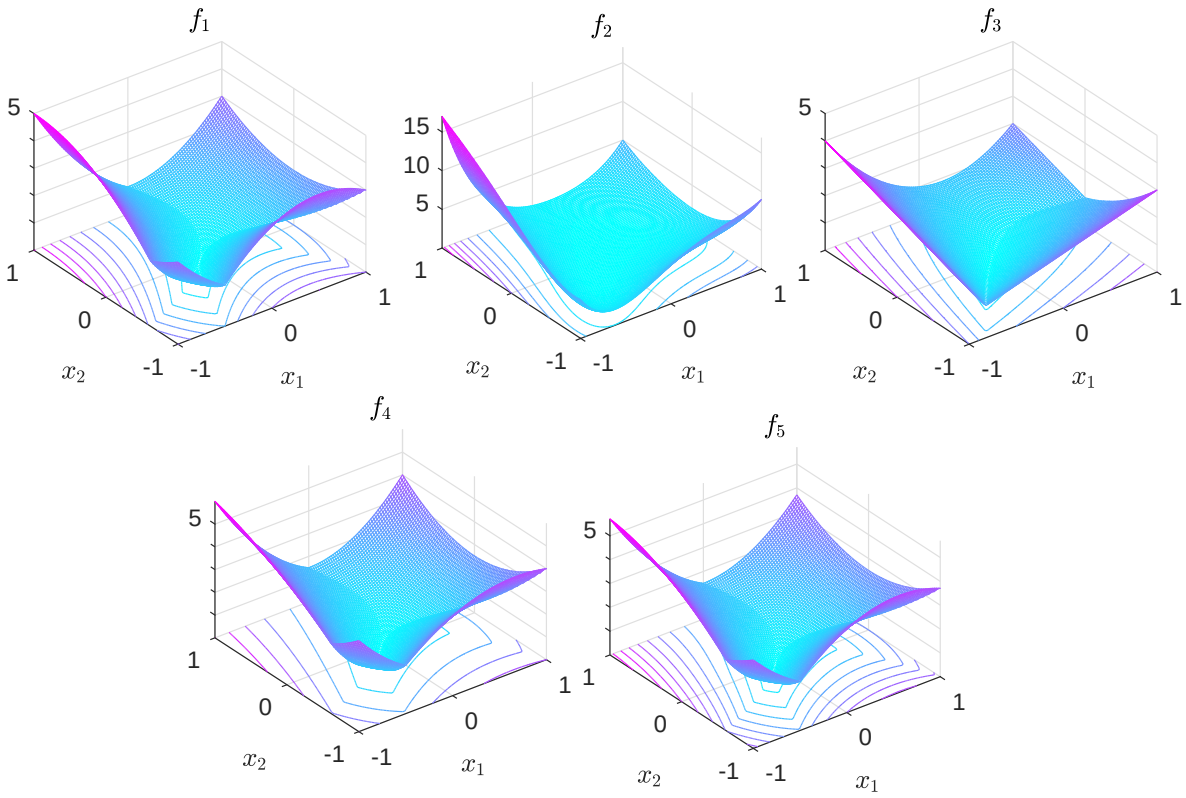

Figure 6: *Graphs of the testfunctions f_1 to f_5 for $x \in \mathbb{R}^2$*

Ferrier polynomials are nonconvex, nonsmooth (except for f_2) and lower-$\mathcal{C}^2$. They all have 0 as a global minimizer [16, p. 23]. The compact constraint set is $X = \{x \in \mathbb{R}^n | |x_i| \leq 10, i = 1, ..., n\}$.

The five test functions f_1 to f_5 are optimized for the dimensions $n = \{2, 3, ..., 15\} \cup \{20, 25, 30, 40, 50\}$. The starting value for each test problem is $x^1 = left(1, \frac{1}{4}, \frac{1}{9}, ..., \frac{1}{n^2} right)^\top$.

For the tests the step size of all algorithms is updated with $\kappa_+ = 1.2$, which provided better results for these specific test functions.

The accuracy measures absolute logarithmic distance to the global minimum. In case that the algorithm finds a local minimum, which is possible for nonconvex objective functions, this lowers the accuracy.

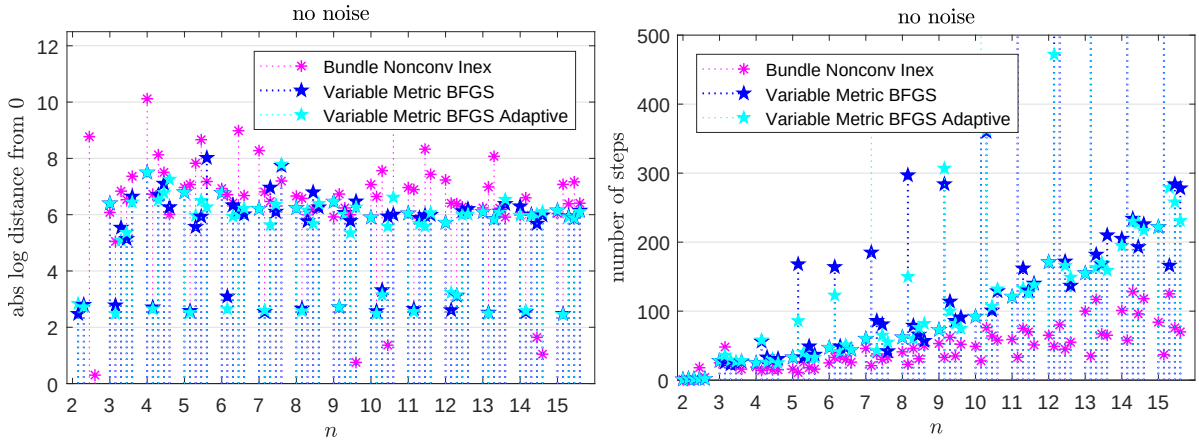

Figure 7: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of no noise.*

In the figures 7 to 9 and 19 to 25 in the appendix the achieved accuracy and the needed number of steps are shown for the proximal bundle method and two versions of the variable metric method.

Figure 7 shows the situation if no noise is present and can be seen as a benchmark for the other noise forms. It is clearly visible that the desired accuracy of 10^{-6} is not always achieved by the different algorithms. A reason for this is that the objective functions have several local minima where the algorithms can get stuck. It seems that this happens more seldom to the proximal bundle algorithm than the variable metric method.

For the Ferrier polynomials the proximal bundle algorithm needs significantly less steps than the variable metric algorithm. It can also be observed that the cases where the variable metric algorithm is stuck in a local minimum, the number of steps needed rises significantly. This is not the case for the proximal bundle algorithm.

The performance of the two variants of the two versions of the variable metric method are similar. It seems however as if the adaptive version performed slightly better in terms of the number of steps used.

In the case of constant noise the variable bundle methods perform better than the proximal version. They are more stable in the achieved accuracy and need considerably less steps than the other method.

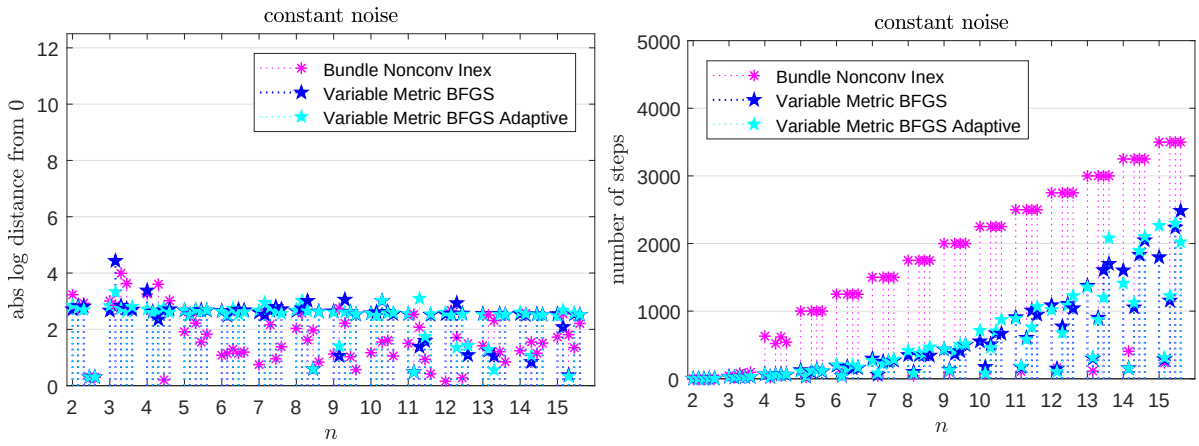

Figure 8: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant noise*

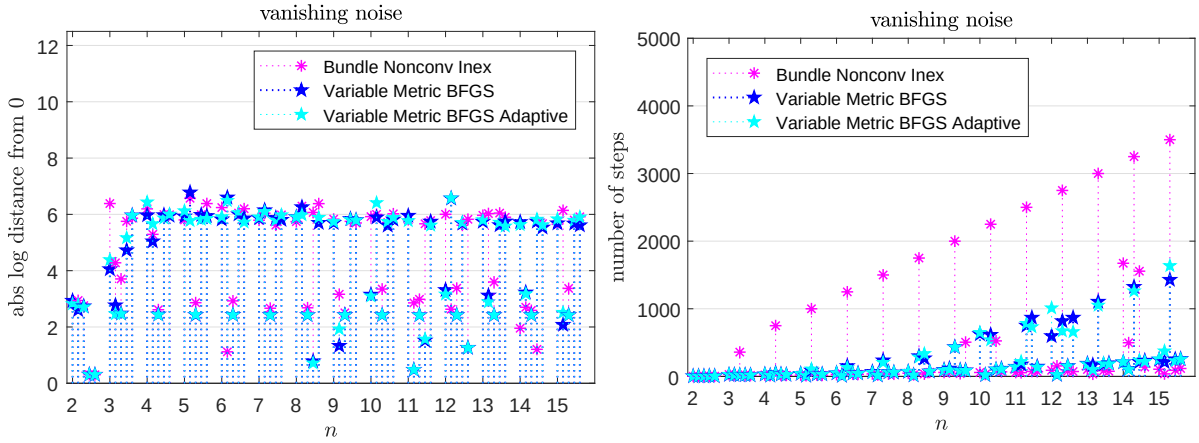

Figure 9: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing noise*

For the other noise forms the three algorithms perform similar in terms of the accuracy but the variable bundle methods need consistently more steps. The only exception from this is case of vanishing noise. Here the proximal bundle method needs extremely many more steps than the variable metric bundle method to optimize function f_3. This shows that the performance of the different algorithms depend on both the form of noise and the specific objective function.

The plots for the higher dimensional data $x \in \mathbb{R}^n$ for $n = \{20, 25, 30, 40, 50\}$ are included in the appendix (figures 21 to 25). Also in higher dimensions the algorithms achieve a similar accuracy. The number of steps needed for convergence in generally higher, but still the proximal bundle method needs less steps in most situations. In the cases

where there is noise on the function value, the algorithms almost always stop because the maximum number of steps is reached. The only exception is the smooth function f_2. Here the variable metric methods perform a lot better than the proximal bundle algorithm in terms of the number of steps, because the curvature information is more reliable.

Finally the influence of the step size updating parameter κ_+ is shown and the performance of the hybrid method. This last method, denoted by 'Variable Metric BFGS, k-scaled' in the figures, uses the scaled BFGS update for the metric matrix Q_k and then scales this matrix again by the step size. This means the final matrix is $Q_k = \frac{1}{k}\tilde{Q}_k$ if $\tilde{Q}_k$ denotes the matrix after the scaled BFGS update, lowering the influence of the metric matrix in each serious step. In this way the method starts out as the variable metric method and then behaves more and more like the proximal bundle method.

This shows that the additional curvature information can make a considerable difference in the convergence speed, especially for matrices Q_k that model the behavior of the objective function at the kinks correctly. It is therefore an interesting but still open question if such matrix updates can be found.

The algorithms used for the comparison of the different κ_+ are endowed with the scaled BFGS update from section 6.4.1. The parameters $\kappa_+ = 1.2$ and $\kappa_+ = 2$ are compared. Here only the exact case and the case of constant noise for the lower dimensions are depicted in figure 10 and 11.

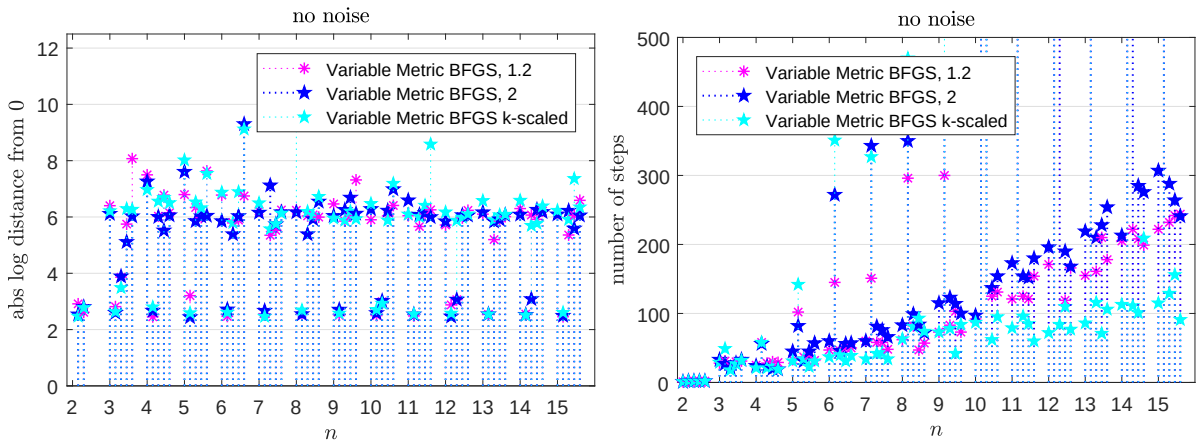

Figure 10: *Influence of the step size updating parameter $\kappa_+ = 1.2$ and $\kappa_+ = 2$ and performance of the hybrid method in the exact case. The reached accuracy is depicted on the left, the needed number of steps on the right.*

One can see that contrary to the academic test example, where the choice $\kappa_+ = 2$ gives better results for the number of steps, here the parameter $\kappa_+ = 1.2$ performs better. In

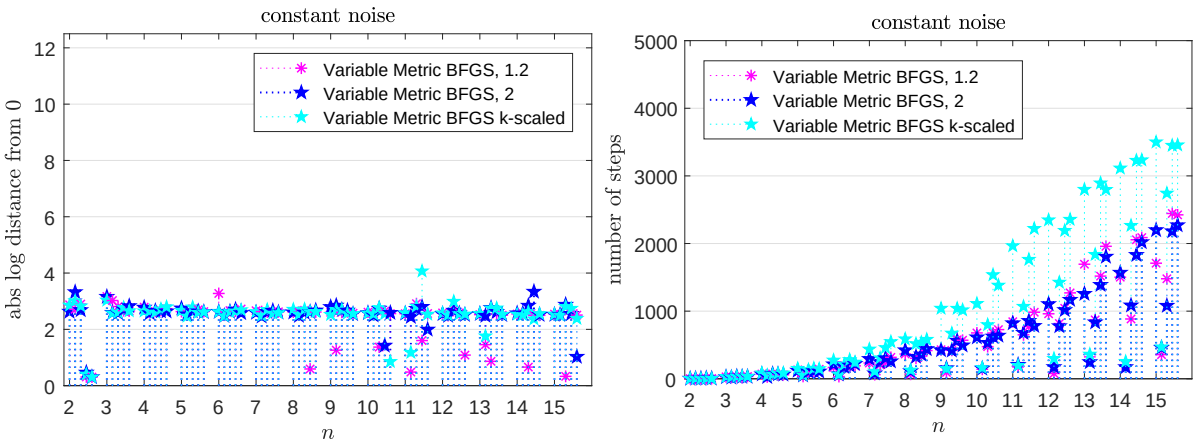

Figure 11: *Influence of the step size updating parameter $\kappa_+ = 1.2$ and $\kappa_+ = 2$ and performance of the hybrid method for constant noise. The reached accuracy is depicted on the left, the needed number of steps on the right.*

the case of constant noise the numbers of steps are similar. As expected the accuracy of the two methods is very similar, because only one parameter is changed. The number of steps shows however that parameter tuning can be useful. Here it is important to keep in mind that the optimal parameter depends on the objective function and the noise form.

The performance of the hybrid method is, as can be expected, similar to the proximal bundle method. As the scaling is quite strong the influence of the metric matrix decreases rather quickly. This means that in cases where the proximal bundle method performed better, the same is true for the hybrid method. The increase in the number of steps is only minor. Likewise in the case of constant noise (Figure 11) for example, where the proximal bundle method needed a lot of steps, these steps are also needed by the hybrid method. Two advantages of the hybrid method still come into play for this noise form: The significant decrease in the number of steps starts only in higher dimensions, where the total number of steps grows larger. Here a 'slower' scaling could yield even better numbers. The other advantage is the more stable accuracy. This holds true for all dimensions.

7. Application to Model Selection for Primal SVM

7.1. Introduction

In this part of the thesis the nonconvex inexact bundle algorithms 5.1 and 6.1 are applied to the problem of model selection for *support vector machines* (SVMs) solving classification tasks. It relies on a bilevel formulation proposed by Kunapuli in [28] and Moore et al. in [41].

A natural application for the inexact bundle algorithm is an optimization problem where the objective function value and the subgradient can only be computed by numerical approximation. This is for example the case in bilevel optimization.

A general bilevel program can be formulated as in [28, p. 20]

$$\begin{aligned} \min_{C \in U_{ad}, \tilde{w} \in \mathbb{R}^k} \quad & \mathcal{L}_{upp}(C, \tilde{w}) && \text{upper level} \\ \text{s.t.} \quad & \mathcal{G}_{upp}(C, \tilde{w}) \leq 0 \\ & \tilde{w} \in \left\{ \begin{array}{cc} \arg\max\limits_{\tilde{w} \in W} & \mathcal{L}_{low}(C, \tilde{w}) \\ \text{s.t.} & \mathcal{G}_{low}(C, \tilde{w}) \leq 0 \end{array} \right\}. && \text{lower level} \end{aligned} \tag{7.1}$$

The two objective functions $\mathcal{L}_{upp}$ and $\mathcal{L}_{low}$ map from $\mathbb{R}^n \times \mathbb{R}^k$ into $\mathbb{R}$ and the constraint functions $\mathcal{G}_{upp}$ and $\mathcal{G}_{low}$ map from $\mathbb{R}^n \times \mathbb{R}^k$ into $\mathbb{R}^l$ and $\mathbb{R}^s$ respectively.

The problem consists of an *upper* or *outer level* which is the overall function to be optimized. Contrary to usual constrained optimization problems which are constrained by explicitly given equalities and inequalities, a bilevel program is additionally constrained by a second optimization problem, the *lower* or *inner level* problem.

Solving bilevel problems can be divided roughly in two classes: implicit and explicit solution methods. In the explicit methods the lower level problem is usually rewritten by its KKT conditions, these are then added as constraints to the upper level problem. Using this solution method the upper and lower level are solved simultaneously. For the setting of model selection for support vector machines as it is used here, this method is described in detail in [28].

The second approach is the implicit one. Here the lower level problem is solved directly in every iteration of the outer optimization algorithm and the solution is plugged into the upper level objective.

Obviously if the inner level problem is solved numerically, the solution cannot be exact. Additionally the *solution map* $S(C) = \{w \in \mathbb{R}^k \mid w \text{ solves the lower level problem for a given } C\}$, can be nondifferentiable [48] and since elements of the solution map are plugged into the outer level objective function in the implicit approach, the outer level function then becomes nonsmooth itself. This is why the inexact bundle algorithm seems a natural choice to tackle these bilevel problems.

Moore et al. use the implicit approach in [41] for support vector regression. However they use a gradient descent method which is not guaranteed to stop at an optimal solution. In [40] it is suggested to use the nonconvex exact bundle algorithm of Fuduli et al. [10] for solving the bilevel regression problem. This method allows for nonsmooth inner problems and can theoretically solve some of the issues of the gradient descent method. It ignores however, that the objective function values can only be calculated approximately; a fact which is not addressed in Fuduli's algorithm.

7.2. Notation

A short remark on the notation in this chapter is required.

Due to standard notation in the field of SVM, the variables x and y are used in this chapter in a different manner than before.

In the setting of SVMs $x^i \in \mathbb{R}^{n_f}$ is an element of the *feature space* that contains the values of one data point. The corresponding variable $y_i \in \{-1, 1\}$ of the *output domain* contains the class in which the data point x^i lies. Sometimes the indices of the samples are omitted to express the general dependency on the data (x,y). The optimization variable of the bundle method is denoted by C during this chapter.

In this chapter there appear also different derivatives. For a continuously differentiable function $f : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ we denote the *gradient* at $(\bar{C}, \bar{\tilde{w}})$ by $\nabla f(\bar{C}, \bar{\tilde{w}}) \in \mathbb{R}^{n+k}$. The partial gradient with respect to $\tilde{w}$ is indicated by $\nabla_{\tilde{w}} f(\bar{C}, \bar{\tilde{w}}) \in \mathbb{R}^k$. For a continuously differentiable vector valued function $F : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^m$ the *Jacobian matrix* is given by $\mathcal{J}F(\bar{C}, \bar{\tilde{w}}) \in \mathbb{R}^{m \times (n+k)}$. The partial Jacobian with respect to the variable $\tilde{w}$ is denoted $\mathcal{J}_{\tilde{w}} F(\bar{C}, \bar{\tilde{w}}) \in \mathbb{R}^{m \times k}$.

Finally let $M \in \mathbb{R}^{n \times k}$ be a matrix and $I \subset \{1, ..., n\}$ an index set. Then the expression M_I denotes the matrix that consists of the rows of M corresponding to the indices in the set I.

7.3. Introduction to Support Vector Machines

Support vector machines are linear learning machines that were developed in the 1990's by Vapnik and co-workers (see [3], where SWMs were introduced). Soon they could outperform several other programs in this area [6] and the subsequent interest in SVMs led to a very versatile application of these machines [28].

The case that is considered here is binary support vector classification using supervised learning. For a thorough introduction to this subject see also [6]. Here a summary of the most important expressions and results is given.

In classification data from a possibly high dimensional vector space $\tilde{X} \subset \mathbb{R}^{n_f}$, the feature or *input space* is divided into two classes. These lie in the output domain $\tilde{Y} = \{-1, 1\}$. Elements from the feature space will mostly be called *data points* here. They get *labels* from the feature space. Labeled data points are called *examples*. The functional relation between the features and the class of an example is given by the usually unknown *response* or *target function* $f(x)$. Supervised learning is a kind of machine learning task where the machine is given examples of input data with associated labels, the so called *training data* (X, Y). Mathematically this can be modeled by assuming that the examples are drawn *identically and independently distributed* (iid) from the fixed joint distribution $P(x, y)$. This usually unknown distribution states the probability that a data point x has the label y [61, p. 988]. The overall goal is then to optimize the generalization ability, meaning the ability to predict the output for unseen data correctly [6, chapter 1.2].

7.3.1. Risk minimization

The concept of SVM's was originally inspired by the statistical learning theory developed by Vapnik. A detailed examination of the subject is given in [60]. In [62] the subject is approached from a more explaining point of view.

The idea of *risk minimization* is to find from a fixed set or class of functions the one that is the best approximation to the response function. This is done by minimizing a loss function that compares the given labels of the examples to the response of the learning machine.

As the response function is not known, only the expected value of the loss can be calculated. It is given by the *risk functional*

$$R(\lambda) = \int \mathcal{L}(y, f_\lambda(x)) \mathrm{d}P(x, y). \tag{7.2}$$

Here $\mathcal{L} : \mathbb{R}^2 \to \mathbb{R}$ is the loss function, $f_\lambda : \mathbb{R}^{n_f} \to \mathbb{R},\ \lambda \in \Lambda$ the approximate response function found by the learning machine and $P(x, y)$ the joint distribution the training data is drawn from. The goal is now to find a function $f_{\bar{\lambda}}(x)$ in a chosen function space $\mathcal{F}$ that minimizes this risk functional [61, p. 989].

As the only given information is provided by the training set inductive principles are used to work with the *empirical risk*, rather than with the risk functional. The empirical risk only depends on the finite training set and is given by

$$R_{\text{emp}}(\lambda) = \frac{1}{n_d} \sum_{i=1}^{n_d} \mathcal{L}(y_i, f_\lambda(x^i)), \tag{7.3}$$

where n_d is the number of data points. The law of large numbers ensures that the empirical risk converges to the risk functional as the number of data points grows to infinity. This however does not guarantee that the function $f_{\bar{\lambda},\text{emp}}$ that minimizes the empirical risk also converges towards the function $f_{\bar{\lambda}}$ that minimizes the risk functional. The theory of consistency provides necessary and sufficient conditions that solve this issue [61, p. 989].

Vapnik therefore introduced the structural risk minimization (SRM) induction principle. It ensures that the used set of functions has a structure that makes it strongly consistent [61]. Additionally it takes the complexity of the function that is used to approximate the target function into account. "The SRM principle actually suggests a tradeoff between the quality of the approximation and the complexity of the approximating function" [61, p. 994]. This reduces the risk of *overfitting*, meaning to overly fit the function to the training data with the result of poor generalization [6, chapter 1.3].

Support vector machines fulfill all conditions of the SRM principle. Due to the kernel trick that allows for nonlinear classification tasks it is also very powerful. For more detailed information on this see [28] and references therein.

7.3.2. Support Vector Machines

In the case of linear binary classification one searches for an affine hyperplane $w \in \mathbb{R}^{n_f}$ shifted by $b \in \mathbb{R}$ to separate the given data. The vector w is called weight vector and b is the bias.

Let the data be linearly separable. This means, that there exists an affine hyperplane (w, b) such that all data points from one class are on the same side of the hyperplane. The function deciding how the data is classified can be written as

$$f(x) = \text{sign}(\langle w, x \rangle - b).$$

Support vector machines aim at finding such a hyperplane that separates also unseen data optimally.

One problem of this approach is that the representation of a hyperplane is not unique. If the plane described by (w, b) separates the data, there exist infinitely many hyperplanes (tw, b), $t > 0$, that separate the data in the same way. To have a unique description of a separating hyperplane the *canonical hyperplane for given data* $x \in X$ is defined by

$$f(x) = \langle w, x \rangle - b \quad \text{s.t.} \quad \min_i \left| \left\langle w, x^i \right\rangle - b \right| = 1.$$

To find such a hyperplane is always possible in the case where the data is linearly separable. It means that the inverse of the norm of the weight vector is equal to the distance of the closest point $x \in X$ to the hyperplane [28, p. 10].

This gives rise to the following definition: The *margin* is the minimal Euclidean distance between a training example x^i and the separating hyperplane. A bigger margin means a lower complexity of the function [6]. It is additionally shown in [28, p. 10] that the margin is proportional to the inverse of $\|w\|$.

A *maximal margin hyperplane* is the hyperplane that realizes the maximal possible margin for a given data set.

Proposition 7.1 ([6, Proposition 6.1]) Given a linearly separable training sample $\Omega = \{(x^1, y_1), ..., (x^{n_d}, y_{n_d})\}$ the hyperplane (w, b) that solves the optimization problem

$$\|w\|^2 \quad \text{s.t.} \quad y_i \left(\left\langle w, x^i \right\rangle - b \right) \geq 1, \quad i = 1, ..., n_d,$$

realizes a maximal margin hyperplane.

The proof is given in [6, chapter 6.1].

Generally one cannot assume the data to be linearly separable. This is why in most applications a so called *soft margin classifier* is used. It introduces the slack variables ξ_i that measure the distance of the misclassified points to the hyperplane:

Fix $\gamma > 0$. A *margin slack variable of the example* (x^i, y_i) with respect to the hyperplane (w, b) and target margin γ is

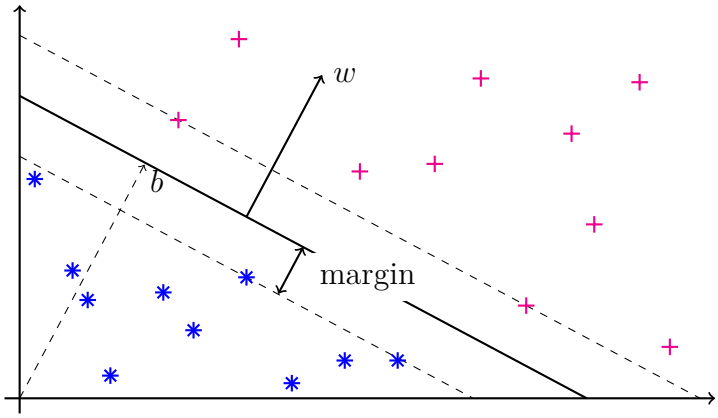

Figure 12: *A separating hyperplane w with bias b. The dashed lines above and below the plane indicate the margin.*

$$\xi_i = \max\left\{0, \gamma - y_i\left(\left\langle w, x^i\right\rangle + b\right)\right\}$$

If $\xi_i > \gamma$ the point is considered misclassified. One can also say that $\|\xi\|$ "measures the amount by which the training set fails to have margin γ" [6, section 2.1.1].

For support vector machines the target margin is set to $\gamma = 1$.

This results in the following optimization problem for finding an optimal separating hyperplane (w, b):

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|_2^2 + \frac{C}{2}\sum_{i=1}^{n_d}\xi_i^2 \\ \text{s.t.} \quad & y_i\left(\left\langle w, x^i\right\rangle - b\right) \geq 1 - \xi_i \\ & \forall i = 1, \ldots, n_d. \end{aligned} \tag{7.4}$$

The first part of the objective function is the regularization, the second part the actual loss function. The parameter $C > 0$ gives a trade-off between the richness of the chosen set of functions f_λ to reduce the error on the training data and the danger of overfitting to have good generalization. It has to be chosen a priori [28].

Instead of the Euclidean norm it is also possible to use the 1-norm in the loss function. Then the resulting optimization problem reads:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n_d}\xi_i \\ \text{s.t.} \quad & y_i\left(\left\langle w, x^i\right\rangle - b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \forall i = 1, \ldots, n_d. \end{aligned} \tag{7.5}$$

In this form, however, the problem does not fit the theory described in [48], which is used later to solve the constructed bilevel problem. In Appendix A.2 it is shown, that for some data sets the strong regularity condition (a crucial concept introduced in section 7.5.1 of this thesis) may not hold in some solutions of this problem. Thus the subgradient of the upper level objective function cannot be calculated as proposed in [48].

In order to fulfill all conditions assumed in [48] it is necessary to reformulate problem (7.4). The new formulation makes use of the *implicit bias*, meaning that the bias is not calculated separately but as part of the variable w. By adding an additional '1' to the end of each feature vector we can use the vectors $\tilde{x}_i := (x_i^\top, 1)^\top$ and $\tilde{w} := (w^\top, w_b)^\top$ to state the following optimization problem:

$$\begin{aligned} \min_{\tilde{w},\xi} \quad & \frac{1}{2}\|\tilde{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n_d}\xi_i^2 \\ \text{s.t.} \quad & y_i\left\langle \tilde{w}, \tilde{x}^i\right\rangle = y_i\left(\left\langle w, x^i\right\rangle - w_b\right) \geq 1 - \xi_i \\ & \forall i = 1, \ldots, n_d. \end{aligned} \tag{7.6}$$

Not treating the bias separately is a strategy used for example to achieve a more efficient implementation [12, section 3.2, p. 22]. In the course of this section the gain of this particular formulation is the fact that it has a unique solution for every $C > 0$ due to strict convexity of the objective function and linearity of the constraints. It is however shown in [12, section 3.2, p. 22] that the solutions that are found with explicit and implicit bias are different. This results from the fact that in case of implicit bias the variable b also enters the regularization term.

From now on we work with the implicit bias. To see the derivation of the bilevel problem with the explicit bias compare for [28, section 2.2].

7.3.3. Multiple Hyper-parameters

To examine the performance of the bilevel approach in the more dimensional case a model suggested by Moore et al. in [41] called *multi-group* support vector classification (multiSVC) is used. This model allows different hyper-parameters for different subgroups of the trainings data. In section 4.3 of [41] the model is described for a regression function. In this thesis the same technique is used for classification.

The motivation behind the approach is that different groups of samples from the training set can have slightly different properties and should therefore have their own weighting parameters C_g. On the one hand this can improve the generalization results. On the other hand properties of the different data groups can be identified by their respective hyper-parameters. Moore et al. explain in [41, section 4.3, p. 9] that for example a large value of C_g signifies reliable data in the respective group whereas a smaller C_g suggests a poorer quality.

To perform multiSVC, divide the trainings data into G (pairwise disjoint) groups. Define the vector of hyper-parameters $C := (C_1, ..., C_G)^\top$. For the sake of simplicity in this thesis all the groups are of equal size but all derivations are also possible for differently sized groups.

The multi-group classification problem in constrained form can be stated as

$$\begin{aligned} \min_{\tilde{w},\xi} \quad & \frac{1}{2}\|\tilde{w}\|^2 + \sum_{g=1}^{G}\left(\frac{C_g}{2}\sum_{i\in N_g}\xi_i^2\right) \\ \text{s.t.} \quad & y_i\left\langle \tilde{w}, \tilde{x}^i\right\rangle \geq 1-\xi_i \\ & \forall i \in \bigcup_{g=1}^{G} N_g = \{1, ..., n_d\}. \end{aligned} \tag{7.7}$$

Here N_g is the index set that contains the indices of the data of the g'th group.

7.4. Formulation of the Bilevel Problem

The hyper-parameters C_g in the objective function of the classification problem have to be set beforehand. This step is part of the model selection process. To set the parameters optimally different methods can be used. A very intuitive and widely used approach is doing *cross validation* (CV) with a grid search implementation [28, p. 30].

To prevent overfitting and get a good parameter selection, especially in case of little data, commonly T-fold cross validation is used [28, p. 30]. For this technique the training data is randomly partitioned into T subsets of equal size. One of these subsets is then left out of the training set and instead used afterwards to get an estimate of the generalization error. To use CV for model selection it has to be embedded into an optimization algorithm over the hyper-parameter space. Commonly this is done by discretizing the parameter space and for T-fold CV training T models at each grid point. The resulting models are then compared to find the best parameters in the grid. Obviously for a growing number of hyper-parameters this is very costly. An additional drawback is that the parameters are only chosen from a finite set [28, p. 30].

A more recent approach is the formulation as a bilevel problem used in [28] and [41]. This makes it possible to optimize the hyper-parameters continuously.

Let $\Omega = \{(x^1, y_1), ..., (x^{n_d}, y_{n_d})\} \subset \mathbb{R}^{n_f+1}$ be a given data set of size $n_d = |\Omega|$. The associated index set is denoted by $\mathcal{N}$. For classification the labels y_i are ± 1. For T-fold cross validation let $\bar{\Omega}_t$ and Ω_t be the training set and the validation set respectively within the t'th fold and $\bar{\mathcal{N}}_t$ and $\mathcal{N}_t$ the respective index sets. For multi-group SVC the index set $\bar{\mathcal{N}}_t$ is again divided into the G sets $\bar{\mathcal{N}}_t^g$ to account for the different groups associated with the different hyper-parameters. Furthermore let $f^t : \mathbb{R}^{n_f+1} \to \mathbb{R}$ be the response function trained on the t'th fold and $\lambda \in \Lambda$ the hyper-parameter to be optimized. For a general machine learning problem with upper and lower loss function $\mathcal{L}_{upp}$ and $\mathcal{L}_{low}$ respectively the bilevel problem reads

$$
\begin{aligned}
&\min_{\lambda, f^t} \quad \mathcal{L}_{upp}\left(\lambda, f^1|_{\Omega_1}, ..., f^T|_{\Omega_T}\right) && \text{upper level}\\
&\text{s.t.} \quad \lambda \in \Lambda \\
&\qquad \text{for } t = 1, ..., T: \\
&\qquad f^t \in \left\{ \begin{array}{ll} \arg\min\limits_{f \in \mathcal{F}} & \mathcal{L}_{low}\left(\lambda, f, (x^i, y_i)_{i=1}^l \in \bar{\Omega}_t\right) \\ \text{s.t.} & \mathcal{G}_{low}(\lambda, f) \leq 0 \end{array} \right\}. && \text{lower level}
\end{aligned}
$$

In the case of multigroup support vector classification the T inner problems have the SVM formulation (7.7). This problem can also be rewritten into an unconstrained form. It is helpful when using the inexact bundle algorithm for solving the bilevel problem. For the t'th fold the resulting hyperplane is identified with the variable $\tilde{w}^t \in \mathbb{R}^{n_f+1}$. The inner level problem for the t'th fold can therefore be stated as

$$\tilde{w}^t \in \arg\min_{\tilde{w}} \left\{ \frac{1}{2}\|\tilde{w}\|^2 + \sum_{g=1}^{G} \left(\frac{C_g}{2} \sum_{i \in \mathcal{N}_t^g} \max\left\{1 - y_i \left\langle \tilde{w}, \tilde{x}^i \right\rangle, 0\right\}^2 \right) \right\}. \tag{7.8}$$

For the upper level objective function different choices are possible. All that are presented here use the implicit bias. They all can also be used with the explicit bias term.

Simply put, the outer level objective should compare the different inner level solutions and pick the best one. An intuitive choice is therefore to pick the misclassification loss, that counts how many data points of the respective validation set Ω_t are misclassified when taking function f^t.

The misclassification loss can be written as

$$\mathcal{L}_{mis}(\tilde{w}) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{N}_t|} \sum_{i \in \mathcal{N}_t} \left(-y_i \left\langle \tilde{w}^t, \tilde{x}^i \right\rangle \right)_\star, \tag{7.9}$$

where the step function $(\cdot)_\star$ is defined component wise for a vector as

$$(r_\star)_i = \begin{cases} 1, & \text{if } r_i > 0, \\ 0, & \text{if } r_i \leq 0 \end{cases}. \tag{7.10}$$

The drawback of this simple loss function is that it is not continuous and thus not suitable for subgradient based optimization. Therefore another loss function is used for the upper level problem, the *hinge loss* or *L1-loss*. It is an upper bound on the misclassification loss and reads

$$\mathcal{L}_{hinge}(\tilde{w}) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{N}_t|} \sum_{i \in \mathcal{N}_t} \max\left\{1 - y_i \left\langle \tilde{w}^t, \tilde{x}^i \right\rangle, 0\right\}. \tag{7.11}$$

It is also possible to square the max term. This results in the *L2-loss* function

$$\mathcal{L}_{hingequad}(\tilde{w}) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{N}_t|} \sum_{i \in \mathcal{N}_t} \max\left\{1 - y_i \left\langle \tilde{w}^t, \tilde{x}^i \right\rangle, 0\right\}^2. \tag{7.12}$$

We refer to this second loss function also as *hingequad loss*.

For the bilevel problem discussed in this thesis the hingequad function is chosen as objective of the upper level problem. Hence the final resulting bilevel formulation for model selection in multigroup support vector classification is

$$
\begin{aligned}
\min_{C} \quad & \mathcal{L}_{hingequad}(\tilde{w}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\max\left\{1-y_i\left\langle\tilde{w}^t,\tilde{x}^i\right\rangle,0\right\}^2 \\
\text{s.t.} \quad & C := (C_1,...,C_G) > 0 \\
& \text{for } t = 1,...,T \\
& \tilde{w}^t \in \arg\min_{\tilde{w}}\left\{\frac{1}{2}\|\tilde{w}\|^2 + \sum_{g=1}^{G}\left(\frac{C_g}{2}\sum_{i\in\mathcal{N}_t^g}\max\left\{1-y_i\left\langle\tilde{w},\tilde{x}^i\right\rangle,0\right\}^2\right)\right\}.
\end{aligned}
\tag{7.13}
$$

Obviously this bilevel problem contains the usual support vector classification with only one data group as a special case.

7.5. Solution with the Inexact Bundle Algorithm

The bilevel problem (7.13) derived above is to be solved with the two bundle algorithms 5.1 and 6.1. This requires having in every iteration an approximate value of the upper level objective function and an approximate subgradient of the upper level objective.

To understand the issues arising when computing especially the subgradient of a bilevel objective consider again the general bilevel problem (7.1).

At the current iterate C^k the function value of the upper level objective function can be computed by solving the lower level problem given C^k. The resulting solution $\tilde{w}^k = (\tilde{w}^{1,k},...,\tilde{w}^{T,k})$ is then inserted into the upper level objective and its value can be calculated.

For computing a subgradient, it is not that simple. Additionally to the variation of the upper level function $\mathcal{L}_{upp}$ with respect to the variable $\tilde{w}$ also the variation of the solution $\tilde{w}^k$ with respect to C has to be considered. To do this we follow the strategy described in [48]. Refer there also for a more thorough analysis on this subject.

7.5.1. Assumptions

Consider the general bilevel problem formulation (7.1) without the explicit constraint $\mathcal{G}_{upp}(C,\tilde{w}) \leq 0$. Let the feasible set U_{ad} be a nonempty compact set and $\tilde{A}$ an open set containing U_{ad}. To use the implicit programming approach suggested in [48], the following assumptions have to hold true:

(A1) The upper level objective $\mathcal{L}_{upp}$ is continuously differentiable on $\tilde{A} \times \mathbb{R}^k$.

(A2) The lower level program possesses a unique solution $\tilde{w}_C$ for every $C \in \tilde{A}$.

(A3) The generalized equation coming from the lower level program is strongly regular at all points $(C, \tilde{w}_C)$, where $C \in \tilde{A}$ and $\tilde{w}_C$ is the corresponding solution of the lower level problem.

A *(perturbed) generalized equation* (GE) is a relation of the form

$$0 \in H(C, \tilde{w}) + N_U(\tilde{w}), \tag{7.14}$$

where $H : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^k$ and $N_U(\tilde{w})$ denotes the *normal cone* to the convex set $U \subset \mathbb{R}^k$ at the point $\tilde{w} \in \mathbb{R}^k$.

Definition 7.2 ([48, Definition 2.6, p. 18]) Let U be a convex subset of $\mathbb{R}^k$ and $\tilde{w}$ in the closure of U. Then

$$N_U(\tilde{w}) := \{\zeta \in \mathbb{R}^k \mid \langle \zeta, v - \tilde{w} \rangle \leq 0 \quad \forall v \in U\}$$

is called normal cone to U at $\tilde{w}$.

The constraint induced by the inner level problem can be rewritten as a generalized equation via its optimality condition. Let U correspond to the feasible set of the inner level problem defined by the intersection of the set W with the set defined by $\mathcal{G}_{low}(\bar{C}, \tilde{w}) \leq 0$ for a fixed variable $\bar{C} \in U_{ad}$. Then the lower level problem can be expressed in an unconstrained way by using the indicator function defined in (3.9). If the constraint set is convex, the indicator function is a convex function and for every element of its subdifferential the subgradient inequality (6.10) holds.

This yields that for all elements ζ of the subdifferential $\partial \mathtt{i}_U(\tilde{w})$ it holds

$$\begin{aligned} &\mathtt{i}_U(v) - \mathtt{i}_U(\tilde{w}) \geq \langle \zeta, v - \tilde{w} \rangle \quad \forall v \in U \\ \Leftrightarrow\, &\langle \zeta, v - \tilde{w} \rangle \leq 0 \quad \forall v \in U, \end{aligned}$$

because for $\tilde{w}, v \in U$ the indicator function is zero. This means that the subdifferential of the indicator function of the set U at the point $\tilde{w}$ coincides with the normal cone to the set U at the point $\tilde{w}$ and so for a constrained minimizer of the function $\mathcal{L}_{low}(C, \cdot)$ on the set U the following optimality condition holds:

$$0 \in \partial \mathcal{L}_{low}(C, \tilde{w}) + N_U(\tilde{w}).$$

If $\mathcal{L}_{low}$ is continuously differentiable, the function $\partial \mathcal{L}_{low} = \{\nabla \mathcal{L}_{low}\} : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^k$ is single valued and continuous and the relation above is a generalized equation.

The strong regularity condition basicly assures, that the GE coming from the lower level problem can provide the needed implicit function.

The following equivalence to this condition is shown in [48]:

Proposition 7.3 ([48, Theorem 5.3, p. 90]) Let the set U be polyhedral. Then the following statements are equivalent:
(i) The GE (7.14) is strongly regular at $(C, \tilde{w})$.
(ii) The map

$$\Lambda : \lambda \mapsto \{\eta \in \mathbb{R}^k \mid \lambda \in \mathcal{J}_{\tilde{w}} H(C, \tilde{w})\eta + N_K(\eta)\}$$

is single valued on $\mathbb{R}^k$. Here K is the critical cone of U, defined in [48, equation (2.50), p. 37].

Let us now verify the above assumptions (A1) – (A3) for the hyper-parameter finding bilevel problem (7.13). First of all, the feasible set U_{ad}, where the hyper-parameter C is chosen from, has to be a convex compact set. In order to assure that, the constraint on the hyper-parameter C has to be adapted. Therefore choose two constants $l_C, u_C > 0$ with $l_C < u_C$ constrain the vector C by the component wise meant inequalities

$$l_C \leq C \leq u_C.$$

Assumption (A1) is fulfilled as $\mathcal{L}_{hingequad}$ is a continuously differentiable function because of the squared max-term.

Next assumptions (A2) and (A3) are verified. In order to do this consider the lower level problem in its constrained formulation (7.7). It can be rewritten in matrix form using the following definitions:

$$H(C) := \begin{pmatrix} \mathbb{I} & \\ & \tilde{C} \end{pmatrix} \in \mathbb{R}^{n_1 \times n_1}$$
$$A := \begin{pmatrix} -y_1(\tilde{x}^1)^\top & -1 & & \\ \vdots & & \ddots & \\ -y_N(\tilde{x}^N)^\top & & & -1 \end{pmatrix} \in \mathbb{R}^{nd \times n_1}, \quad a := \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix} \in \mathbb{R}^{nd}.$$

Here $\tilde{C}$ is a diagonal matrix with the vector $(C_1, ..., C_1, ..., C_G, ..., C_G)^\top$ on the diagonal such that the hyper-parameters correspond to the ξ_i of the different groups. The number $n_1 = n_f + 1 + n_d$ where n_f is the dimension of the feature space and n_d the number of data points in the sample.

Then the lower level problem can be written as

$$\begin{aligned} \min_{\tilde{w},\xi} \quad & \frac{1}{2}(\tilde{w}^\top, \xi^\top) H(C) \begin{pmatrix} \tilde{w} \\ \xi \end{pmatrix} \\ \text{s.t.} \quad & A \begin{pmatrix} \tilde{w} \\ \xi \end{pmatrix} \leq a. \end{aligned} \tag{7.15}$$

This formulation of the lower level problem corresponds to the one in Appendix A.1.3 in [48]. Only the objective function depends on the hyper-parameter C, not the constraints.

Problem (7.15) has a strictly convex objective function and linear constraints. Therefore it possesses a unique global minimizer. Denoting the vector of all slack variables ξ_i with ξ this minimizer is given by $((\tilde{w}^*)^\top, (\xi^*)^\top)^\top$. It depends on the vector of hyper-parameters C.

Due to convexity, for this problem the KKT conditions are necessary and sufficient. This means that at the minimum $((\tilde{w}^*)^\top, (\xi^*)^\top)^\top$ holds

there is $\mu^* \in \mathbb{R}^{nd}$ such that
$$\nabla_{(\tilde{w},\xi)} L(C, \tilde{w}^*, \xi^*, \mu^*) = 0$$
$$\mu^* \geq 0, \quad A \begin{pmatrix} \tilde{w}^* \\ \xi^* \end{pmatrix} \leq a, \quad \left\langle \mu^*, \left(A \begin{pmatrix} \tilde{w}^* \\ \xi^* \end{pmatrix} - a \right) \right\rangle = 0,$$

for a given C, where $L(C, \tilde{w}, \xi, \mu) = \frac{1}{2}(\tilde{w}^\top, \xi^\top)H(C)(\tilde{w}^\top, \xi^\top)^\top + \left\langle \mu, \left(A(\tilde{w}^\top, \xi^\top)^\top - a\right)\right\rangle$ is the Lagrangian of the problem and μ^* the multiplier corresponding to the inequality constraints.

Using the KKT conditions a special GE can be derived for problem (7.15):

$$0 \in \begin{bmatrix} \nabla_{(\tilde{w},\xi)} L(C, \tilde{w}, \xi, \mu) \\ -A\left(\tilde{w}^\top, \xi^\top\right)^\top + a \end{bmatrix} + N_{\mathbb{R}^{n_f+1} \times \mathbb{R}^{n_d}_+}. \tag{7.16}$$

The derivation of this particular form of a GE is given in [48, chapter 4, p. 71–72 and chapter 5, p. 92].

This new GE is very convenient due to the simple structure of the cone but it depends now also on the Lagrange multiplier μ. Therefore this multiplier also has to be unique at the solution $((\tilde{w}^*)^\top, (\xi^*)^\top)^\top$. To check this introduce the following index sets:

$$\begin{aligned} I(\tilde{w}, \xi) &= \{i \in \{1, ..., n_d\} \mid A(\tilde{w}^\top, \xi^\top)^\top - a = 0\} \\ I_+(\tilde{w}, \xi) &= \{i \in I(\tilde{w}, \xi) \mid \mu_i > 0\}, \end{aligned}$$

which denote the set of *active* and *stronly acitve* inequality constraints respectively. The set I_+ also depends on the multiplier μ. In cases where μ is not unique, this has to be indicated by writing $I_+(\tilde{w}, \xi, \mu)$. As it is shown in the following that for the presented problem the multiplier μ is always unique, it is however omitted in the specification of I_+.

To assure that the optimal Lagrange multiplier μ^* is unique, we show that the *linear independence constraint qualification* (LICQ) holds in every minimum $((\tilde{w}^*)^\top, (\xi^*)^\top)^\top$ [48, Theorem 4.8, p. 82], [64, Theorem 1, p. 3].

In the present case, where there are only inequality constraints, the LICQ holds at a point $(\tilde{w}, \xi)$ if the rows of the matrix $\mathcal{J}_{(\tilde{w},\xi)}\left(A(\tilde{w}^\top, \xi^\top)^\top - a\right) = A$ that correspond to the indices in $I(\tilde{w}, \xi)$ are linearly independent [48, p. 82, 96].

For problem (7.15) the LICQ holds at every feasible point $(\tilde{w}, \xi)$ because all rows of the matrix A are linearly independent. This means that linear independence particularly holds for all rows corresponding to the active constraints at any minimum $(\tilde{w}^*, \xi^*)$. Hence the lower level problem possesses a unique solution vector $(\tilde{w}^*, \xi^*)$ and a unique corresponding Lagrange multiplier μ^* for every hyper-parameter (vector) C. Thus assumption

(A2) is fulfilled.

Finally to show that assumption (A3) holds, strong regularity has to be shown for the generalized equation (7.16). This is done by using Theorem 5.8 in [48, p. 96].

For stating the theorem the following definition is needed:

Definition 7.4 ([48, Definition 4.4, p. 81]) Let $M \in \mathbb{R}^{n\times n}$ be a matrix and $K \subset \mathbb{R}^n$ a cone. The matrix M is strictly copositive with respect to K if

$$\langle d, Md\rangle > 0 \quad \text{for all} \quad d \in K, d \neq 0.$$

Theorem 7.5 (c.f. [48, Theorem 5.8, p. 96]) *Suppose that LICQ holds at* $(\tilde{w}, \xi)$ *and that the matrix* $\mathcal{J}_{(\tilde{w},\xi)}\left(\nabla_{(\tilde{w},\xi)} L(C, \tilde{w}, \xi, \mu)\right)$ *is strictly copositive with respect to the kernel of the matrix* A_{I_+}.
Then the GE (7.16) is strongly regular at $(C, \tilde{w}, \xi, \mu)$.

To assure that the Hessian of the Lagrangian is well defined, the lower level objective function has to be two times continuously differentiable. This is given for the quadratic objective function of (7.15) and it holds that

$$\mathcal{J}_{(\tilde{w},\xi)}\left(\nabla_{(\tilde{w},\xi)} L(C, \tilde{w}, \xi, \mu)\right) = \begin{pmatrix} \mathbb{I} & \\ & \tilde{C} \end{pmatrix}.$$

Because all components of the vector $(C_1, ..., C_1, ..., C_G, ..., C_G)^\top$ are greater than zero, this matrix is positive definite and as such strictly copositive on the whole $\mathbb{R}^{n_f+1+n_d}$. This means that all necessary conditions are fulfilled in order to compute a subgradient with respect to C of the upper level objective function $\mathcal{L}_{upp}(C, \tilde{w}(C))$ in the way presented in [48].

7.5.2. The Adjoint Problem

In order to calculate the needed subgradient of bilevel problem the technique of *adjoint equations* is used. This technique is well known from optimal control (c.f. [48, p. 126] and references therein). For the derivation of the adjoint problem and the results used below, refer to chapters 6 and 7 of [48].

The adjoint problem to the given bilevel problem (7.13) can be stated as the following constrained quadratic problem using the expressions defined in (7.5.1):

$$\begin{aligned} \min_{p} \quad & \frac{1}{2}\langle p, H(C)p\rangle - \left\langle \nabla_{(\tilde{w},\xi)}\mathcal{L}_{upp}(\tilde{w},\xi), p\right\rangle \\ \text{s.t.} \quad & A_{I_+M}p = 0. \end{aligned} \tag{7.17}$$

Here the index set $I_+M = I_+(\tilde{w},\xi) \cup M_i(\tilde{w},\xi)$. The set $M_i(\tilde{w},\xi) \subset I(\tilde{w},\xi) \setminus I_+(\tilde{w},\xi)$ is a subset of the inequality constraints, that are active but not strictly active at the point $(\tilde{w},\xi)$ and has to be chosen suitably. What this means is explained next.

First introduce the notation $I_0(\tilde{w},\xi) = I(\tilde{w},\xi) \setminus I_+(\tilde{w},\xi)$. Constraints whose indices are in this set are also calles *weakly active*. Let $\mathcal{P}(I_0(\tilde{w},\xi))$ be the power set of this index set and $\mathbb{K}(\tilde{w},\xi)$ an index set that contains indices for all elements of the set $\mathcal{P}(I_0(\tilde{w},\xi))$. The index set $M_i(\tilde{w},\xi)$ is then the element of $\mathcal{P}(I_0(\tilde{w},\xi))$ corresponding to the index $i \in \mathbb{K}(\tilde{w},\xi)$. To ensure a proper choice of the index set $M_i(\tilde{w},\xi)$ [48] offers a theorem and a corollary that are applicable in the given context.

For better readability the brackets behind the index sets are omitted. All index sets refer to the same point.

Theorem 7.6 (c.f. [48, Theorem 7.10, p. 137]) *Consider the GE (7.16) at the point $(C,\tilde{w},\xi,\mu)$. Suppose that the assumptions (A2) and (A3) hold and that for a given $i \in \mathbb{K}(\tilde{w},\xi)$ the following system in (z_1,z_2,z_3,z_4) is inconsistent:*

$$\begin{aligned} -(A_{I_0\setminus M_i})^\top z_1 + (H(C))^\top z_3 - (A_{I_+\cup M})^\top z_4 &= 0 \\ z_2 + A_{M_i} z_3 &= 0 \\ \left(\mathcal{J}_C\left(H(C)(\tilde{w},\xi)^\top\right)\right)^\top z_3 &= 0 \\ (z_1,z_2) \geq 0, \quad (z_1,z_2) &\neq 0 \\ z_3 \in \ker(A_{I_+}). & \end{aligned}$$

Then subgradient can be calculated via the method above.

Corrolary 7.7 (c.f. [48, Corollary 7.12, p. 139]) *Let the assumptions of Theorem 7.6 be fulfilled. Suppose that the constraints do not depend on the variable C and that*

$$\ker\left(\mathcal{J}_C\left(H(C)(\tilde{w}^\top,\xi^\top)^\top\right)^\top\right) \cap \ker\left(A_{I_+}\right) = \{0\}.$$

Then the subgradient can be calculated via the method above for each $i \in \mathbb{K}(\tilde{w},\xi)$.

Neither the assumptions of the corollary nor those of the theorem can be assured in general. They have therefore to be checked for the specific situation in every iteration. For the specific instances used for the numerical experiments in this thesis there never existed any weakly active constraints. So in this thesis the set $M_i(\tilde{w}, \xi)$ was always chosen empty.

7.6. On Error Bounds and Regularity

In order to use algorithms 5.1 and 6.1, some properties are required of the objective function and the nature of the inexactness. In this section we give a short comment on how far theses properties can be met.

7.6.1. Error Bounds

It is assumed in (5.2) that the error on the function value and subgradient are bounded. This bound does not have to be known, but it has to exist and allows an estimation of the possible accuracy to which the algorithms can solve the given problem (c.f. Lemma 5.2).

Due to the complicated structure of bilevel problems no tight error bounds can be given for problem (7.13) but the existence of general error bounds can be shown.

The error on the function value in bilevel problems originates from the fact, that in the implicit approach the numerically calculated minimal point of the lower level is inserted into the upper level function. Of course, this optimal point can only be approximated to a given tolerance. As the lower level problem of (7.13) is a strictly convex quadratic optimization problem, a very common stopping criterion is to check for approximate first order optimality.

For constrained problems first order optimality means that the KKT conditions (which are necessary and sufficient in the convex case) are fulfilled. Approximate optimality then means, that the conditions hold true within the bounds of a chosen stopping tolerance $\texttt{tol} > 0$.

The solver that is used for the numerical solution of the problem in this thesis is the `quadprog` solver from MATLAB. This solver uses the above stopping condition scaled by the input values [37].

This however cannot give an estimate for the minimizing argument of the lower level objective. This can be seen when taking a strictly convex whose slope is so small, that

the stopping condition is fulfilled for every point on a compact subset of the feasible set U_C. In this case also the approximate minimizing point found by any algorithm can be any point in U_C.

It still can be assured, that the minimizer $\tilde{w}$ found by an algorithm is not infinitely far away from the exact minimizer $\tilde{w}^*$. This follows immediately if the constraint set is compact. If it is unbounded in any direction this follows from strict convexity. Without loss of generality instead of the KKT conditions one can consider the first order optimality condition in the unconstrained case, namely that the norm of the gradient at a minimizer is zero. (Due to unboundedness of the constraint set in the examined directions, the problem can be treated as an unconstrained one for them. In the direction where the constraint set is bounded, possible choices for $\tilde{w}$ are also bounded.) For strictly convex functions the gradient is strictly monotone, hence the area where its norm is smaller or equal to any chosen tolerance is compact. This means that in this case all possible choices of $\tilde{w}$ are bounded.

Using local Lipschitz continuity of the upper level objective function on the admissible set we can conclude that

$$\|\mathcal{L}_{upp}(\tilde{w}) - \mathcal{L}_{upp}(\tilde{w}^*)\| \leq L_{upp}\|\tilde{w} - \tilde{w}^*\| \leq L_{upp} D_{\tilde{w}}(\texttt{tol}),$$

where L_{upp} is the Lipschitz constant of the upper level objective on the admissible set. The number $D_{\tilde{w}}(\texttt{tol})$ is the maximal distance of points $\tilde{w}_1$ and $\tilde{w}_2$ in the compact set originating from the intersection of the constraint set and the set where the lower level objective has a gradient smaller than the chosen tolerance `tol`.

An equally loose bound can be provided for the approximate subgradient via local Lipschitz continuity of the upper level objective and Lipschitz continuity of the solution map [48, chapter 5]. The inexactness of the subgradient has two reasons. On the one hand the subgradient is not calculated at the exact minimizer $\tilde{w}^*$ but at an approximated point $\tilde{w}$. As the subdifferential is bounded by the Lipschitz constant on U_{ad} the bound for that part of the subgradient error that originates from taking the subgradient only at an approximate point is given by $2L_{upp}L_w$. Here $L_{\tilde{w}}$ is the Lipschitz constant of the solution function $\tilde{w}(C)$.

On the other hand for computation of the subgradient the adjoint problem (7.17) has to be solved numerically and therefore only gives an approximate solution. As the adjoint problem is also a strictly convex constrained optimization problem its minimizing argument p can be estimated the same way as above.

Hence the distance between the exact subgradient and the approximate one in the sense of p is

$$\begin{aligned}\langle \mathcal{J}_C H(C)\tilde{w}^*, (p^* - p)\rangle &\leq \|\mathcal{J}_C H(C)\|_2 \|\tilde{w}^*\| \|p^* - p\| \\ &\leq \|\max\{1, C_{max}\}\| \tilde{w}_{max} D_p(\texttt{tol}).\end{aligned}$$

Here C_{max} denotes the maximum component of the vector of hyper-parameters and $\tilde{w}_{max}$ is the maximum of the function $\tilde{w}(C)$ on U_{ad}. The sum of those bounds is the overall bound of the approximate subgradient error.

7.6.2. Regularity

For the proof of Lemma 5.2 subdifferential regularity of the upper level objective function with the solution map inserted is required.

We have however to remark here, that from Proposition 7.4 in [48] only follows weak semismoothness of this function. In [55, p. 82] an example is given that semismooth, quasidifferentiable function does not have to be subdifferentially regular. As semismooth functions are a subset of weakly semismooth functions, we cannot conclude that the objective function of (7.13) with the solution of the lower level plugged in is regular.

Proving regularity for the particular instance (7.13) of a bilevel problem may be possible, taking into account the structure provided by the strictly convex objective functions of both levels and the fact that the class of semismooth and regular functions is identical to the class of lower-$\mathcal{C}^1$ functions [16, equation (3), p. 5]. It lies however beyond the scope of this thesis.

7.7. Numerical Experiments

In this section the two bundle algorithms 5.1 and 6.1 are applied to solve the classification tasks described above. The algorithms are compared to the MATLAB routines `fminsearch` and `fminbnd` in the one dimensional case and to `fmincon` for the multiGroup application.

7.7.1. The Data

There are five data sets used. Two of them are real world data sets available on the UCI Machine Learning repository [34] and three synthetic data sets. The synthetic data sets were particularly designed for the optimization of the hyper-parameter C.

A correct choice of the hyper-parameter is particularly important in the case were the data of the different folds is especially sensitive to overfitting. This is the case in the following situation: The data of each of the different folds is nearly separable, but only with a rather small margin. Additionally, the optimal hyperplanes for the different folds have different directions from the optimal hyperplane, because they are only fitted to minimize the error on the training set.

This situation is shown in the figures below for the synthetic data set `synsmall`.

It can be seen, that the data sets of the different folds are all linearly separable, but only for very specific hyperplanes with a small margin. These hyperplanes can be found by leaving the regularization term $\frac{1}{2}\|\tilde{w}\|$ out of the objective function. It can also be seen, that the optimal hyperplane found if C is chosen optimally, is similar to only two of the dashed lines. The line found for the second fold is very different to those.

The situation shown in the plots can also be expressed in numbers. Remember that the margin is proportional to the inverse of the norm of the vector w. The matrix containing the vectors for the different folds is given as

$$W = \begin{pmatrix} 116.7223 & 10.2238 & 26.9331 \\ -103.4875 & -19.8813 & -26.6014 \end{pmatrix}$$

and the bias as $b = (20.2872, 0.7818, 0.679)$.

One can already see that the norms of the vectors are rather large: The norms are $\text{norm}(w^1) = 155.9928$, $\text{norm}(w^2) = 22.3560$ and $\text{norm}(w^1) = 37.8554$ for $W = (w^1, w^2, w^3)$. Compare this to the optimal plane (w, b): Here $w = (2.7842, -2.9033)$, $b = 0.2239$ and $\text{norm}(w) = 4.0226$.

For linear classification, the danger of overfitting is rather small. This is also indicated by the fact that from a group of different real world data sets, only the two used here provide better results with the regularization term than without it. However, the chance of overfitting is higher the richer the class of functions is, that is used to separate the two classes [28, p. 2-4]. This means, that in case of nonlinear classification overfitting can become a problem. There is also a third situation where overfitting can occur more

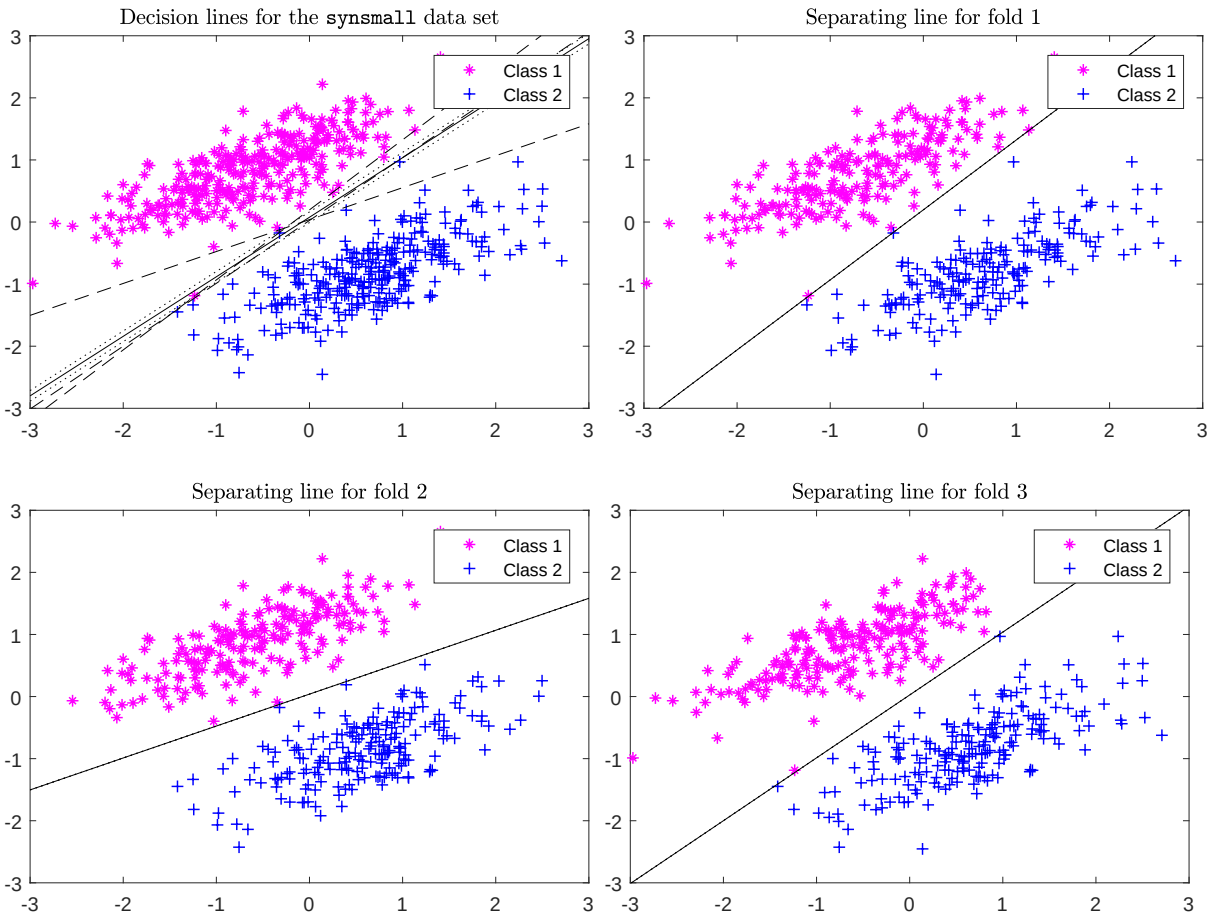

Figure 13: *The figure on the top left shows the* `synsmall` *data set. The solid line is the optimal decision line, with the margin indicated as dotted lines on either side of it. The unregularized decision boundaries for the single folds are drawn with dashed lines.*
The other three plots show the training data corresponding to the first to third fold of the `synsmall` *set and the corresponding unregularized decision lines.*

often. This is when the data set consists of relatively little data compared to the number of features. In this case overfitting can happen more often already for linear classification, because the high dimensional space in which the hyperplane lies offers more degrees of freedom to fit the plane to the data.

The two real world data sets used in this thesis are the Breast Cancer Wisconsin data set and Johns Hopkins University Ionosphere database. We refer to them as `cancer` and `ionosphere`, respectively. The three two dimensional synthetic data sets are called `synbox`, `synsmall` and `synbig`.

For the `synbox` data set, two groups of data were uniformly distributed over two boxes of size 4×2. For the four different folds the boxes are located differently: In the first fold,

the boxes are positioned directly above each other, without any overlapping. This means that the data in this fold is linearly separable. For the second to fourth fold, the upper box is gradually pushed down 'into' the lower box, such that the data is not linearly separable any more.

For the `synsmall` and the `synbig` data sets, the two groups are normally distributed in two dimensions with mean being $\mu^1 = (2,3)^\top$for the first group and $\mu^2 = (-1,9)^\top$ for the second group. The covariance is given by

$$\Sigma = \begin{pmatrix} 1.5 & 1.5 \\ 1.5 & 3 \end{pmatrix}$$

for both groups. The data set `synbig` is a very large data set to test the performance of the different algorithm when handling such data sets. For both sets the different folds are chosen in a way that they are separable but the separating hyperplane does not give a good generalization.

All data sets were standardized to have a zero mean and unit standard deviation. Then the data sets were split up into three (four) folds of equal size. It is shown in [28, p. 47] that this is a reasonable number of folds. From the two real world data sets, a part of the data is hold back for validation purposes (in table 2 marked with n_v). The real world data sets were randomly put into the folds. For the synthetic data sets the folds were selected as described above. The number of instances and number of features of every data set is given in Table 2.

Data set	n_d	n_v	n_f	T
`cancer`	240	443	9	3
`ionosphere`	240	111	33	3
`synbox`	600		2	4
`synsmall`	600		2	3
`synbig`	30000		2	3

Table 2: *Properties of the used data sets for one-dimensional optimization.*

7.7.2. Choice of Parameters

The bilevel problem (7.13) is solved with the two presented bundle algorithms 5.1 and 6.1. For bilevel problems it is not possible to evaluate the upper level objective without solving the lower level problem first. Also additional steps are necessary to calculate a subgradient.

For the numerical calculations done in this sections, the bundle algorithms therefore work after the following algorithmic pattern, which is inserted into the respective routines described in chapters 5 and 6.

Algorithmic Pattern 7.1

Initialize the algorithm.
For $k = 1, 2, 3, \ldots$

1. Calculate the step d^k by solving bundle subproblem (5.16) or (6.3) and the new iterate $C^{k+1} = C^k + d^k$.
2. Solve the lower level of bilevel problem (7.13) with a QP solver for C^{k+1}.
3. Calculate a subgradient of the upper level function by solving the adjoint problem (7.17).
4. Calculate the aggregate objects, update all necessary variables and test for a serious step as given in the respective bundle algorithm.
5. STOP if a stopping condition is matched.

The bundle algorithms are set up with the following parameter values: The descent parameter m is set to 0.05, $\gamma = 2$, the initial step size is $t_0 = 0.1$ and the step size updating parameters are $\kappa_- = 0.8$ and $\kappa_+ = 1.2$. Elements are added to the bundle if the corresponding Lagrange multiplier $\alpha_j^k > 10^{-15}$. For method 6.1 the scaled BFGS update is used with the threshold being $q = 10^8$. Both algorithms stop if either the decrease measure δ_k is smaller than a certain tolerance or if a maximum number of steps of 1000 is reached. The bundle subproblems are solved to a tolerance of 10^{-15} or stop after 5000 iterations. They are solved with MATLAB's `quadprog`. Finally the overall optimization variable C is constrained component wise to $10^{-5} < C < 10^4$.

The bundle methods 5.1 and 6.1 are compared to different MATLAB routines. These only work with the function values and do not need to have a derivative or subgradient provided by the user. For every evaluation of the upper level objective function first the T lower level problems are solved. Then the resulting hyperplanes $\tilde{w}^t$, $t = 1, ..., T$ are plugged into the upper level objective to calculate the loss.

All calculations were done on an Intel i5 2.6 GHz with four kernels.

7.7.3. One-dimensional Optimization

First the algorithms are tested for the one-dimensional case. Here the bundle methods are compared to the `fminbnd` algorithm `fminsearch`. Both can solve one-dimensional optimization problems. It has to be remarked here, that the `fminsearch` routine does not accept any bounds on the optimization variable. Strictly speaking it could thus give negative values for C, which are infeasible. This did however not happen in any of the experiments.

In the one dimensional case it is possible to plot the objective function values of the bilevel problem. The plots are given in Figure 14.

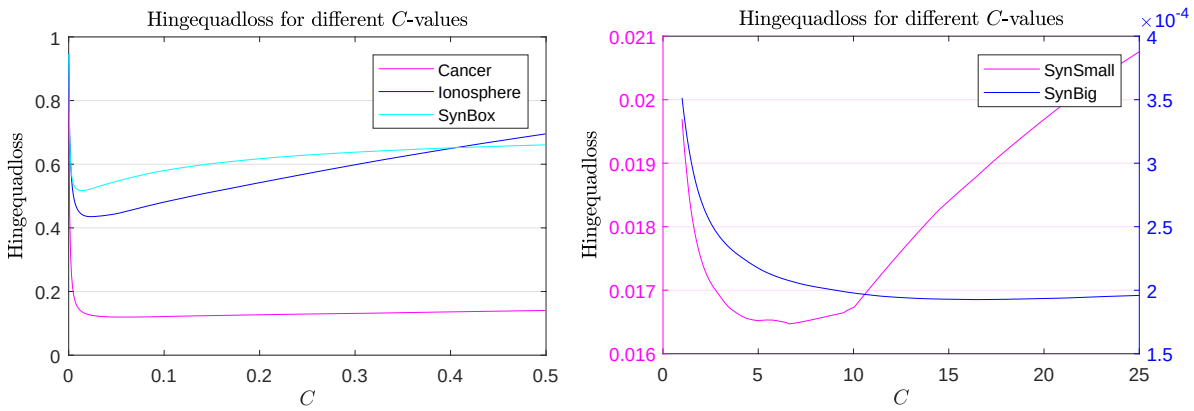

Figure 14: *Plots of the hingquad error for different C-values values. The figure on the left shows the plot for the* `cancer`, `ionosphere` *and* `syn box` *data sets. The plot on the right depicts the situation for the sets* `synsmall` *(left axis) and* `synbig` *(right axis).*

Different aspects of the algorithms are compared for various choices of the starting value, scaling of the objective function, stopping tolerances and exactness of the subroutines.

At first the behavior of the different algorithms regarding differently scaled upper level objective functions and the sensitivity of the different methods towards the starting value are analyzed.

The starting values C_0 are taken from the set $\{10^{-5}, 0.01, 1, 20, 100\}$. The `fminbnd` algorithm does not need any starting value. In Figure 14 it can be seen, that especially for the sets `synsmall` and `synbig` the objective functions are very flat. In order to increase the accuracy, the objective functions are scaled in the following experiment. We compare the unscaled version of the hingequadloss and a scaled version with scaling factor 100. The stopping tolerance of all methods is set to `tol` $= 10^{-6}$.

For this comparison the lower level problem as well as the adjoint problem of the bundle methods are solved with a tolerance of 10^{-15} for the constraints and optimality condition. Both problems are solved with the MATLAB QP solver `quadprog` which stops after meeting the tolerance or after a limit of 1000 iterations is reached.

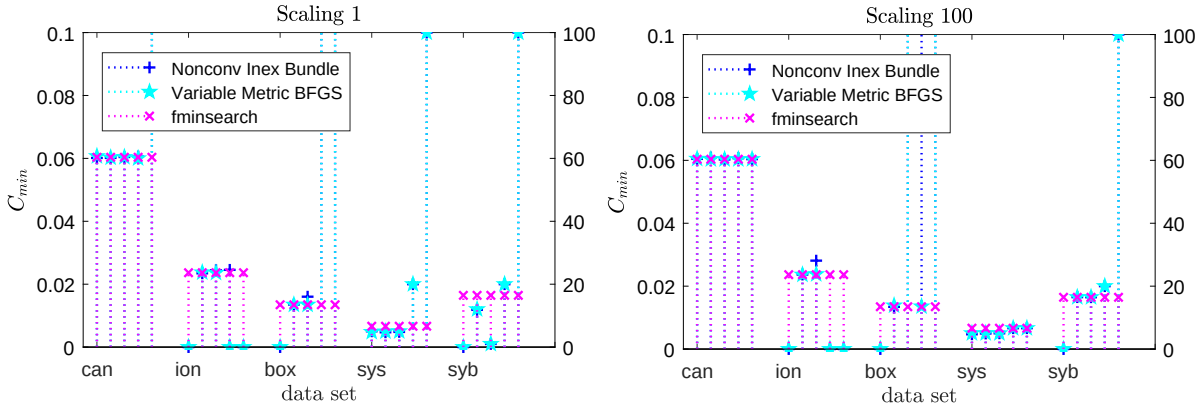

Figure 15: *The plots show the minimizers found by* `fminseach` *and the two bundle methods for different starting values on the different data sets. For each data set, the lines next to each other show the result for the starting values* $C_0 = (10^{-5}, 0.01, 1, 20, 100)$.
The Plot on the left shows the situation for the unscaled objective function. For the plot on the right the objective is scaled by 100.
The first three data sets are measured with the scale on the left, the other two data sets with the scale on the right.

In figures 15 and 16 the minimizers and function values found for the different starting values and scaling factors are depicted for the different solution methods. It can be observed from the plots, that `fminsearch` finds the same minimizers for all starting values and both scaling functions. For the two bundle methods, it depends on the starting value, whether a good minimizer is computed. For the unscaled version of the objective function, the bundle methods often stop without any calculations, reporting the starting value as optimal. This happens more often for the larger starting values 20 and 100. For the `synsmall` data set it depends on the starting value, weather the local or the global minimum is found.

For the starting value $C_0 = 1$ and the scaling factor 100 decent results are achieved. Hence for this parameter combination the results of all four methods used, are presented more accurately in table 3.

One can see that the MATLAB methods `fminsearch` and `fminbnd` solve the problem very accurately.The routine `fminsearch` arrives at the same minimizers as `fminbnd`. Sometimes it needs however more computation time. This is particularly notable for the data

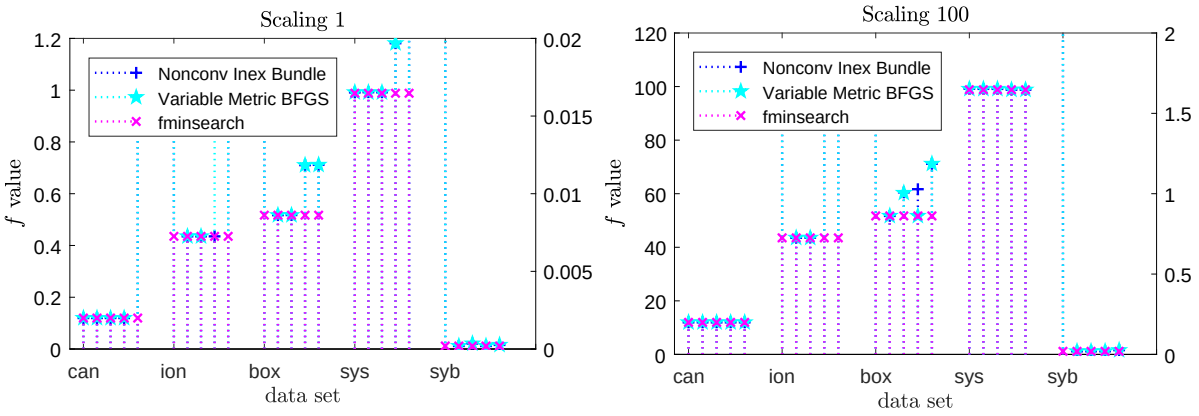

Figure 16: *The plots show the minimal value found by* `fminseach` *and the two bundle methods for different starting values on the different data sets. For each data set, the lines next to each other show the result for the starting values* $C_0 = (10^{-5}, 0.01, 1, 20, 100)$.
The Plot on the left shows the situation for the unscaled objective function. For the plot on the right the objective is scaled by 100.
The first three data sets are measured with the scale on the left, the other two data sets with the scale on the right.

set `synbig`, where `fminbnd` needs less than half of the solution time of `fminsearch`. Both algorithm reach the minimizers that can be identified from the plots to an accuracy of about 10^{-6}. For the `synsmall` data set they find the global minimum, not the local one.

Both bundle algorithms perform rather badly compared to the MATLAB routines. Although the bundle algorithms can use subgradient information which the MATLAB routines lack. It can be observed, that for the `synsmall` data set, the bundle methods solve the task faster, but the solutions are not as exact as the ones obtained by the MATLAB solvers. For the `synbig` data set, all algorithms need considerably more time solving the optimization problem than for the other data sets. A reason for this is, that the lower level problem needs more solution time due to its increased size. As also the adjoint problem increases in size (the constraints of both, the lower level and the adjoint problem, scale with the number of data points) the bundle methods need more time solving the bilevel program. It can be seen, that the approach to find the subgradient via the adjoint problem is not appropriate for larger data sets.

The bundle algorithms 5.1 and 6.1 show similar results in terms of computation time. Here the second order information used by algorithm 6.1 has no measurable influence. Both algorithms reach an accuracy of about 10^{-3} of the minimizer compared to the MATLAB routines. The accuracy of the optimal found function value lies around 10^{-4} for the objective scaled with 100 if a correct minimizer is found.

Data set		Bundle 5.1	Bundle 6.1	`fminsearch/bnd`
`cancer`	C_{min}	0.0603	0.0603	0.0604
	f value	11.9644	11.9644	11.9644
	time (s)	5	13	4/2
`ionosphere`	C_{min}	0.0281	0.0238	0.0236
	f value	43.5899	43.5242	43.5241
	time (s)	2284	1729	6/30
`synbox`	C_{min}	0.1592	0.1535	0.0135
	f value	60.4899	60.2966	51.7078
	time (s)	255	331	26/549
`synsmall`	C_{min}	5.0112	5.0228	6.6372
	f value	1.6520	1.6520	1.6470
	time (s)	4	6	11/7
`synbig`	C_{min}	16.4743	16.4965	16.4595
	f value	0.0193	0.0193	0.0193
	time (s)	593	595	673/306

Table 3: *This table shows the minimizer, corresponding minimal function value and the needed computation time of the four tested routines* `fminsearch`*,* `fminbnd`*, algorithm 5.1 and 6.1. The starting value is* $C_0 = 1$ *and the used objective function is scaled with 100.*

The convexification parameter η_k is assumed to stay bounded in [16, p. 11]. Although all of the objective functions have a "generally convex" form, the maximum η_k occurring in the algorithms is of order 10^6. Generally η_k seems to reach higher values for the scaled objective function as it can be seen in Table 3.

It is now tested, if a lower stopping tolerance of the bundle methods can increase their accuracy. For this experiment only the starting value $C_0 = 1$ and the scaled objective function are taken. The lower level and adjoint problems are still solved to an accuracy of 10^{-15}. The accuracy of the overall method is increased from 10^{-6} to 10^{-10}.

Interestingly it can be observed, that for algorithm 5.1 the minimizer that is found by the algorithm only changes for the `synbig` data set. For all other sets, the exact same results are found. This is not the case for method 6.1. Here better results are found for the data sets `synbox`, `synsmall` and `synbig`. However the effective accuracy of the function value does not change compared to the case when the stopping tolerance is 10^{-6}. The extra computation time needed for the more exact case is comparatively low. We conclude from that, that for most of the data sets the calculations of both methods (5.1 and 6.1) are already much below the tolerance of 10^{-6} when the algorithms stop. This means, that no, or only a few more calculations are needed to meet also a smaller stopping tolerance. All in all these insights suggest, that it is not possible, to achieve as exact solutions of

the bilevel problems as the MATLAB procedures do with the bundle algorithms, even if the stopping tolerance is lowered.

Finally the performance of the methods for different levels of accuracy is tested. To do this, algorithms 5.1 and 6.1 run with the lower level and adjoint problem only solved inaccurately. The different tolerances that are used are 10^{-4} and 10^{-8}

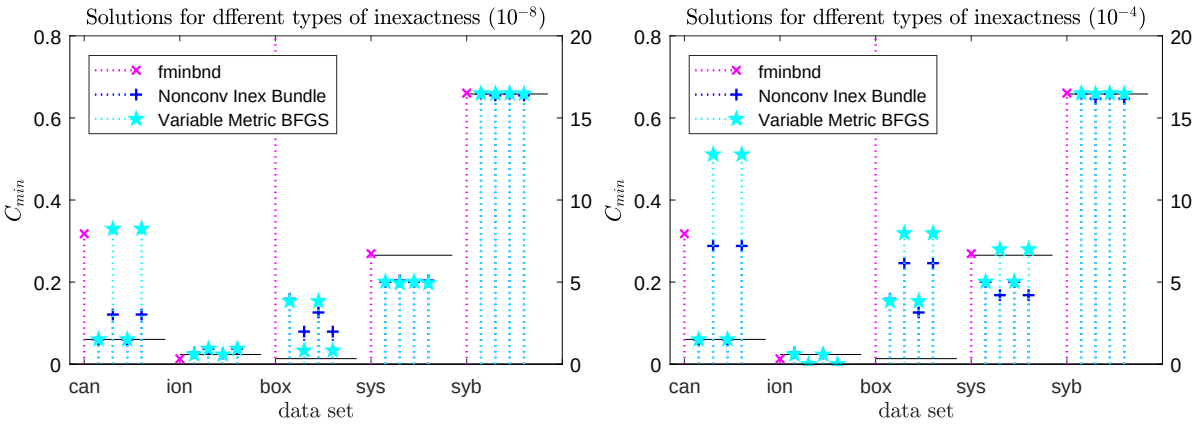

Figure 17: *The Plots show the minimizers that are found for different stopping tolerances of the adjoint and lower level program.*
The black bar over each set shows the result that is achieved with the `fminbnd` *method if the lower level problem is solved to a tolerance of* 10^{-15}*. The pink stem shows the result of* `fminbnd` *for a tolerance of the lower level of* 10^{-8} *in the left plot and* 10^{-4} *in the right plot. The four following stems for each data set show the results of the combinations (from left to right): Lower level and adjoint problem solved for tolerance* 10^{-15}*; lower level solved for* 10^{15}*, adjoint for* $10^{-8}, 10^{-4}$*; lower level solved for* $10^{-8}, 10^{-4}$ *adjoint for* 10^{-15}*; lower level and adjoint problem solved for* $10^{-8}, 10^{-4}$
The first three data sets are measured with the scale on the left, the other two data sets with the scale on the right.

The other parameters are set in a way that achieved good results in the comparison before. Precisely this means that the objective function is scaled by 100 and the bundle methods start with value $C_0 = 1$. The found minimizers and minimal function values for the different combination of tolerances are depicted in figures 17 and 18 respectively.

What can be seen is, that the solution tolerance of the adjoint problem does not show any influence on the accuracy of the bundle algorithms. The minimizer and the function value basically stay the same, no matter if the adjoint problem is solved for an accuracy of 10^{-4}, 10^{-8} or 10^{-15}. The reason for this is, that the adjoint problem calculates the subgradient. It could already be seen in section 6.5.2, that inexactness of the subgradient has a relatively small influence on the overall solution.

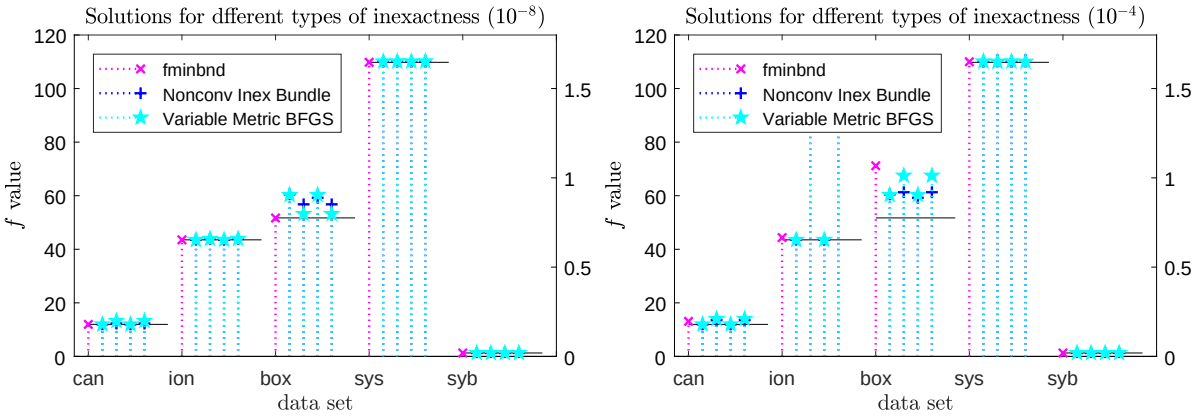

Figure 18: *The Plots show the minimial function values that are found for different stopping tolerances of the adjoint and lower level program.*
The black bar over each set shows the result that is achieved with the `fminbnd` *method if the lower level problem is solved to a tolerance of* 10^{-15}*. The pink stem shows the result of* `fminbnd` *for a tolerance of the lower level of* 10^{-8} *in the left plot and* 10^{-4} *in the right plot. The four following stems for each data set show the results of the combinations (from left to right): Lower level and adjoint problem solved for tolerance* 10^{-15}*; lower level solved for* 10^{15}*, adjoint for* $10^{-8}, 10^{-4}$*; lower level solved for* $10^{-8}, 10^{-4}$ *adjoint for* 10^{-15}*; lower level and adjoint problem solved for* $10^{-8}, 10^{-4}$*.*
The first three data sets are measured with the scale on the left, the other two data sets with the scale on the right.

The plots also show, that the solution accuracy of the lower level can have big influence on the value of the minimizer. This is also true for the results found by `fminbnd`. This results from the little change in the objective function values. It can be observed, that although the value of the minimizer is sometimes far away from the correct one, the resulting function value is still near the optimum.

The final step of this analysis of the one-dimensional case is to study how much the 'worse' results of the bundle methods actually influence the generalization ability of the optimal model. This is tested by calculating the misclassification error on the validation sets of the `cancer` and `ionosphere` data. Therefore, an optimal separating hyperplane $\tilde{w}$ is calculated, given the optimal hyper-parameter C_{min}. For this calculation all training data is used. The hyper-parameter has to be scaled by $\frac{T}{T-1}, T > 1$ [28, p. 47] for the use with only one fold. As mentioned before, we use the misclassification loss as given in (7.9) to calculate the fraction of the data that is misclassified.

One can see, that the SVM problem itself is not too sensitive to the choice of the hyper-parameter. This means, that despite the fact, that the MATLAB routines provide better results in numbers, the bundle methods find an equally good hyperplane.

Data set	Bundle 5.1	Bundle 6.1	`fminbnd`
`cancer`	0.038374	0.038374	0.038374
`ionosphere`	0.153153	0.162162	0.162162

Table 4: *Misclassification error on the validation data. The separating hyperplane is generated using the C_{min} value found by the respective optimization algorithm.*

To summarize this section it can be said, that in the one-dimensional case, the MATLAB routines outperform the bundle methods. The algorithm `fminbnd` finds very accurate results, often in less time than the bundle methods. Additionally it does not need a starting value. We can conclude, that in the one-dimensional case, the bundle algorithms work for solving the presented problem, but there are better options available.

7.7.4. Multi-dimensional Optimization

In the multigroup approach, the data set is partitioned into different groups, that are each weighted with their own parameter C_g. To calculate the optimal value of all these parameter results in a multi-dimensional optimization problem.

In this section the bundle algorithms 5.1 and 6.1 are compared to the matlab method `fmincon`. This method is suitable for the solution of a very wide range of constrained multi-dimensional optimization problems. They can be nonlinear and do not have to be convex.

The following data sets are used for the comparison:

Data set	n_d	n_f	T	G
`cancer1`	960	9	3	4
`cancer2`	678	9	3	2
`cancer3`	678	9	3	2
`cancer4`	675	9	3	3
`ionospere1`	960	33	3	4
`ionosphere2`	348	33	3	2

Table 5: *Properties of the data sets for multi-dimensional optimization.*

They are constructed in the following ways: The sets `cancer1`, and `ionosphere1` are constructed by duplicating the original data sets G and adding different amounts of uniformly distributed noise on the data of the different groups. The sets `cancer2`, `cancer3`, `cancer4` and `ionosphere2` are constructed by using also the validation data of the original data sets and partitioning it randomly in G groups of equal size. For the `cancer2`

data set, there is additionally noise added to the second group.

First the multigroup bilevel problem is solved by the three algorithms `fmincon`, 5.1 and 6.1 for all data sets. The objective function is again scaled by 100. The stopping conditions are meeting the stopping tolerance `tol` $= 10^{-6}$ and exceeding the maximal number of steps (1000). All subproblems are calculated to a tolerance of 10^{-10}. The starting values for the different dimensions are set to $C_{0,2} = (5, 0.5)^\top$, $C_{0,3} = (5, 0.5, 0.05)^\top$ and $C_{0,4} = (5, 0.5, 0.05, 1)^\top$.

Table 6 show the results for the different algorithms.

It can be seen, that again the MATLAB routine `fmincon` reaches the lowest function values. However often the computation time is also longer than the ones of the two bundle methods. An interesting fact is, that the various found minimizers are very different from each other. This might be due to the flat shape of the objective functions, which can be observed in the two-dimensional case. However, scaling the objective function with higher numbers gives even worse results for the bundle methods. For the sets `cancer1`, `cancer3` and `ionosphere1` it can not be detected, that the groups with the added noise are weighted consistently lower, than the exact group.

The function values decrease for the bundle methods if the subproblem is solved to a higher accuracy (here 10^{-15}). For `fmincon` this effect is barely measurable. However, the computation time is about doubled for `fmincon` in that case, whereas it does not rise for the bundle algorithms. This indicates, that in the nearly exact case, the bundle algorithm may profit from the subgradient information. Like in the one-dimensional case, it does not influence the results if the adjoint problem is computed to a lower tolerance. The same holds true if the stopping tolerance is lowered. This is as expected, since the same behavior can also be seen in the one-dimensional case.

We remark that the parameter η_k reaches high values up to 10^8 for both bundle methods.

As in the one-dimensional case we close this analysis by comparing how well the generated models work on unseen data. Therefore we compute the optimal hyperplanes, given the optimal hyper-parameters C_{min}, on the whole set of training data. Then this hyperplane is used to classify the validation set and the numbers of misclassified points are compared. The hyper-parameters are scaled in the same way as above to compensate for using only one fold.

Again it can be observed, that the variance in the hyper-parameter does not result in an equally board variance of the misclassification error.

All in all, we conclude that again both bundle algorithms, 5.1 and 6.1, are able to solve

the given bilevel problem. The minimizers, that are found by the various algorithms, are very different. Still the computed loss function values are close together. The `fmincon` routine is able to generate the lowest function values. The bundle algorithms can compare to the computation time of the MATLAB algorithm and for very accurately calculated inner level problems, they outperform `fmincon` in terms of computation time.

Data set		Bundle 5.1	Bundle 6.1	`fmincon`
`cancer1`	C_{min}	$\begin{pmatrix} 0.1525 \\ 1.01 \cdot 10^{-5} \\ 10^{-5} \\ 4.5948 \end{pmatrix}$	$\begin{pmatrix} 0.1642 \\ 10^{-5} \\ 10^{-5} \\ 5.5829 \end{pmatrix}$	$\begin{pmatrix} 0.3149 \\ 0.5369 \\ 1.3596 \\ 8.22 \cdot 10^{3} \end{pmatrix}$
	f value	49.7412	49.7381	49.7105
	time (s)	23	16	146
`cancer2`	C_{min}	$\begin{pmatrix} 10^{-5} \\ 0.0421 \end{pmatrix}$	$\begin{pmatrix} 0.0714 \\ 1.06 \cdot 10^{-4} \end{pmatrix}$	$\begin{pmatrix} 0.0693 \\ 0.0259 \end{pmatrix}$
	f value	10.7188	10.2587	10.1394
	time (s)	5	7	9
`cancer3`	C_{min}	$\begin{pmatrix} 0.0389 \\ 0.0573 \end{pmatrix}$	$\begin{pmatrix} 0.0550 \\ 0.0165 \end{pmatrix}$	$\begin{pmatrix} 0.0555 \\ 0.0386 \end{pmatrix}$
	f value	11.3871	11.4340	11.3438
	time (s)	71	9	15
`cancer4`	C_{min}	$\begin{pmatrix} 0.0476 \\ 0.2342 \\ 0.1778 \end{pmatrix}$	$\begin{pmatrix} 0.0219 \\ 0.0829 \\ 0.0679 \end{pmatrix}$	$\begin{pmatrix} 0.0202 \\ 0.0771 \\ 0.0642 \end{pmatrix}$
	f value	9.8094	9.6319	9.6308
	time (s)	134	27	16
`ionosphere1`	C_{min}	$\begin{pmatrix} 0.0078 \\ 0.0169 \\ 0.0282 \\ 10^{-5} \end{pmatrix}$	$\begin{pmatrix} 10^{-5} \\ 0.0478 \\ 10^{-5} \\ 10^{-5} \end{pmatrix}$	$\begin{pmatrix} 0.0074 \\ 0.0455 \\ 10^{-5} \\ 10^{-5} \end{pmatrix}$
	f value	44.1661	46.6613	44.0434
	time (s)	246	7	59
`ionosphere2`	C_{min}	$\begin{pmatrix} 0.0116 \\ 0.0189 \end{pmatrix}$	$\begin{pmatrix} 10^{-5} \\ 0.0215 \end{pmatrix}$	$\begin{pmatrix} 0.0116 \\ 0.0189 \end{pmatrix}$
	f value	47.3260	49.7165	47.3260
	time (s)	10	4	9

Table 6: *This table shows the minimizer, corresponding minimal function value and the needed computation time of the three tested routines* `fmincon`*, algorithm 5.1 and 6.1.*
The used objective function is scaled with 100.

Data set	Bundle 5.1	Bundle 6	`fmincon`
`cancer1`	0.038374	0.038374	0.036117
`cancer2`	0.027088	0.033860	0.031602
`cancer3`	0.029345	0.029345	0.029345
`cancer4`	0.027088	0.029345	0.0293453
`ionosphere1`	0.198198	0.216216	0.198198
`ionosphere2`	0.144144	0.135135	0.144144

Table 7: *Misclassification error on the validation data. The separating hyperplane is generated using the C_{min} value found by the respective optimization algorithm.*

8. Conclusion

In this thesis, bundle methods, that are suitable to solve optimization problems with non-smooth, nonconvex objective functions and inexact function and subgradient information, are investigated.

In chapter 4 it was investigated how different bundle methods tackle inexactness and nonconvexity. With the help of these insights, the bundle algorithm 5.1 was analyzed for possible simplifications and extensions. Different special cases were stated under which the convergence proof provides stronger results. Additionally it was possible to show that the aggregation technique works with the presented algorithm.

In chapter 6 algorithm 5.1 was extended to use also approximate second order information of the objective function. A convergence proof of the method was provided and the new method 6.1 was compared to algorithm 5.1 for different updating procedures. We found, that both methods perform rather similar. For some academic test examples, the second order information could enhance the performance of algorithm 6.1.

In the last chapter the former two bundle methods were used to solve model selection problem for support vector classification. The emerging optimization problems could be solved by both bundle methods. However, it seems that for the calculation strategies used in this thesis, other algorithms can provide more accurate results. We concluded, that the assets of the presented bundle methods lie in different fields of applications.

A. Appendix

A.1. Omitted Proofs

In this section the proofs that were omitted in the main part of the thesis are given.

A.1.1. Eigenvalues of the Metric Matrix

Proposition A.1 Let $A \in \mathbb{R}^{n\times n}$ a symmetric matrix, $b \in \mathbb{R}$ and $\mathbb{I} \in \mathbb{R}^{n\times n}$ the identity matrix. Let λ_i^A, $i = 1, ..., n$ be the eigenvalues of the matrix A. Then the eigenvalues of the matrix $A + b\mathbb{I}$ are given by $\tilde{\lambda}_i := \lambda_i^A + b$ for all $i = 1, ..., n$.

Proof: Due to the symmetry of the matrix A it possesses n real eigenvalues λ_i^A. Let v^i, $i = 1, ..., n$ be the corresponding eigenvectors. It follows that for $i = 1, ..., n$

$$(A + b\mathbb{I})v^i = Av^i + bv^i = (\lambda_i^A + b)v^i.$$

This means that v^i is an eigenvector of $A+b\mathbb{I}$ to the eigenvalue λ_i^A+b for all $i = 1, ..., n$. □

A.1.2. Proof of Proposition 6.2

Proof: We show that the scalar product $\langle x, y\rangle_{Q_k+\frac{1}{t_k}\mathbb{I}} := x^\top(Q_k + \frac{1}{t_k}\mathbb{I})y$ is well-defined. This yields directly that also the norm induced by the scalar product is well-defined (see for example [35, Corollary 12.6, p. 172]).

By proposition 6.1 the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ is bounded and relation (6.6) assures that the following calculations are valid for all k.

We prove now that the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ can be used to define a scalar product.

From the rules for matrix-vector multiplication follows that

$$(x + y)^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z = x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z + y^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z, \quad x, y, z \in \mathbb{R}^n$$

and

$$x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) (y + z) = x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) y + x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z, \quad x, y, z \in \mathbb{R}^n.$$

Thus linearity of the defined scalar product is proven.

The symmetry of $Q_k + \frac{1}{t_k}\mathbb{I}$ yields symmetry of the scalar product by

$$x^\top \underbrace{\left(Q_k + \frac{1}{t_k}\mathbb{I}\right) y}_{:=\tilde{y}} = \tilde{y}^\top x = y^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right)^\top x = y^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) x.$$

Finally positive definiteness of the scalar product follows directly from positive definiteness of the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$.

This means the scalar product $\langle \cdot, \cdot \rangle_{Q_k + \frac{1}{t_k}\mathbb{I}}$ is well defined and thus induces the norm $\| \cdot \|_{Q_k + \frac{1}{t_k}\mathbb{I}}$. □

A.2. Counterexample to Strong Regularity

In section 7.3.2 of this thesis it is claimed that problem (7.5) may not be strongly regular at its solution.

It is remarked in [48, p. 96] that if LICQ does not hold for a minimum (w^{top}, b, ξ^{top}), then the GE coming from this problem is not strongly regular in this point.

In order to show that a situation where LICQ is violated in the minimum can be easily constructed, consider $w \in \mathbb{R}^{n_f}$. In a usable machine learning problem the number of data points n_d is much larger than the size n_f of the feature space, in order to achieve reliable generalization.

Consider now, that $n_f + 2$ of the data points, that correspond to active constraints are correctly classified and sit directly on the margin. (This means, they are so called *support vectors*. Refer to section 6.1.1 of [6] for more information on this.) For those points both of the constraints

$$y_i \left(\left\langle w, x^i \right\rangle - b \right) \geq 1 - \xi_i \quad \text{and} \xi_i \geq 0$$

hold with equality.

This means, that by renumbering these constraints the matrix with the derivatives of them has the following form:

$$\tilde{A} = \begin{pmatrix} -y_1(x^1)^\top & 1 & -1 & & \\ \vdots & \vdots & & \ddots & \\ -y_{n_f+2}(x^{n_f+2})^\top & 1 & & & -1 \\ 0 & 0 & -1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & -1 \end{pmatrix}.$$

As it was assumed that there are $n_f + 2$ such data points, this matrix cannot have linearly independent rows. This means that LICQ does not hold in the given solution and therefore also strong regularity is violated.

A.3. Additional Figures

A.3.1. Variable Metric Bundle Method

The following plots show the behavior in accuracy and number of steps of the proximal bundle algorithm 5.1 and different realizations of the variable metric bundle method 6.1 when optimizing the Ferrier polynomials f_1 to f_5 in different dimensions and for different noise forms. The conditions and parameters used for the plots are described in section 6.5.2

The two plots below depict the situation for $x \in \mathbb{R}^n$ for $n = 2, 3, ..., 15$.

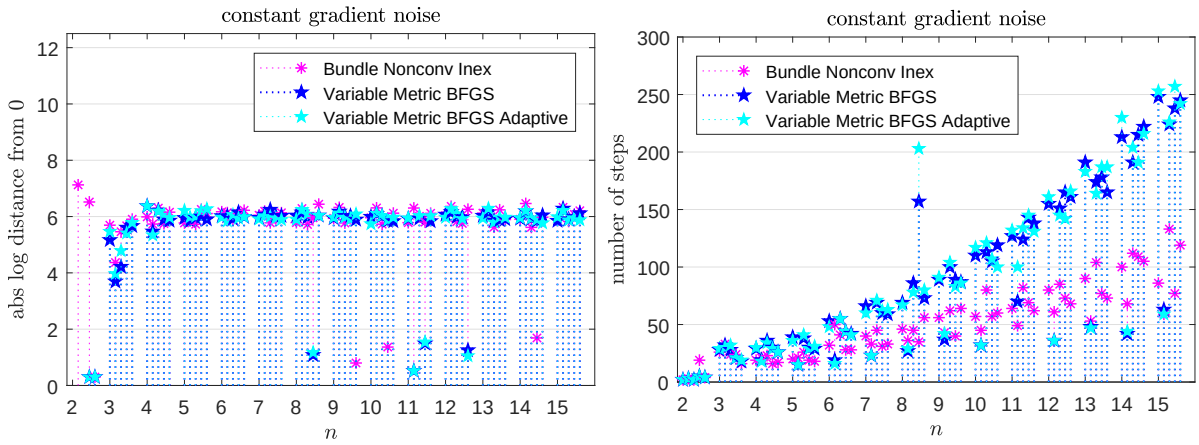

Figure 19: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant gradient noise*

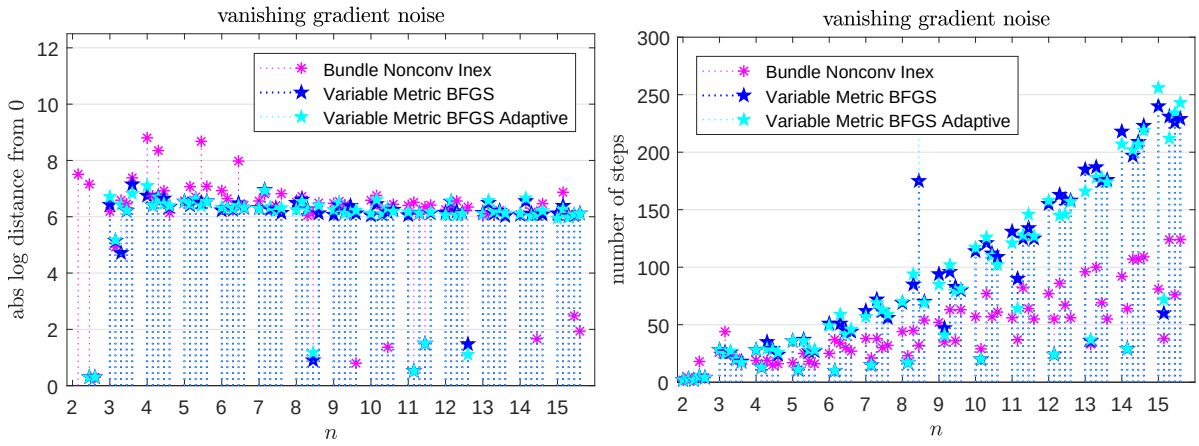

Figure 20: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing gradient noise*

The following plots show the situation for larger dimensions $n = \{20, 25, 30, 40, 50\}$.

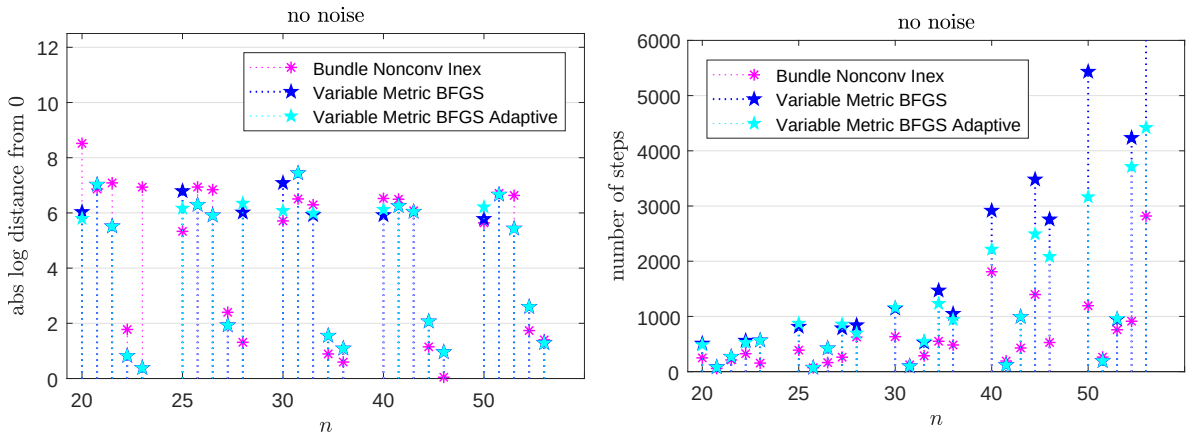

Figure 21: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of no noise*

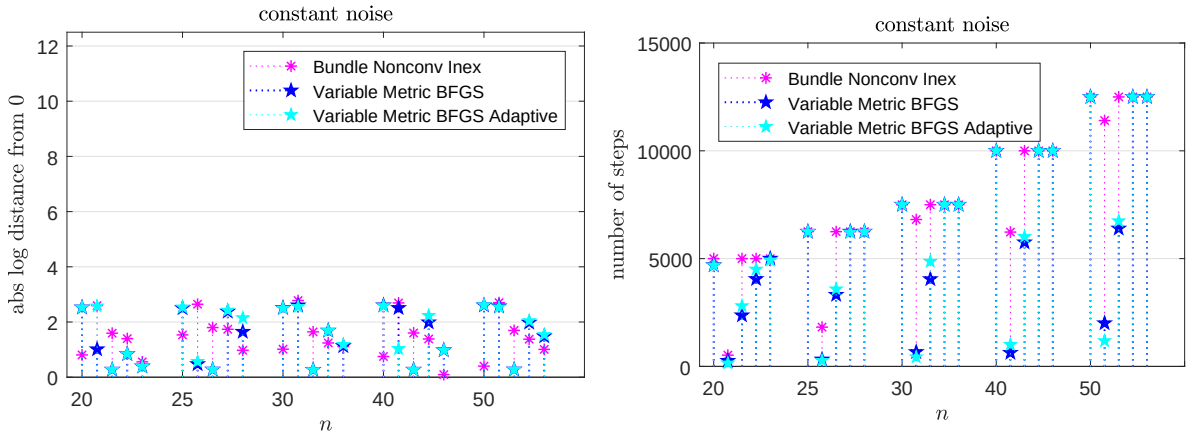

Figure 22: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant noise*

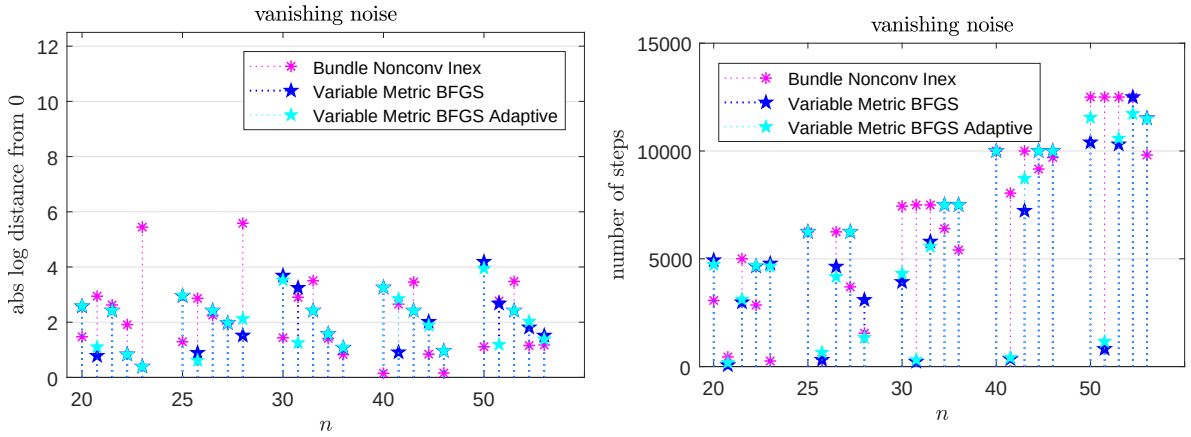

Figure 23: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing noise*

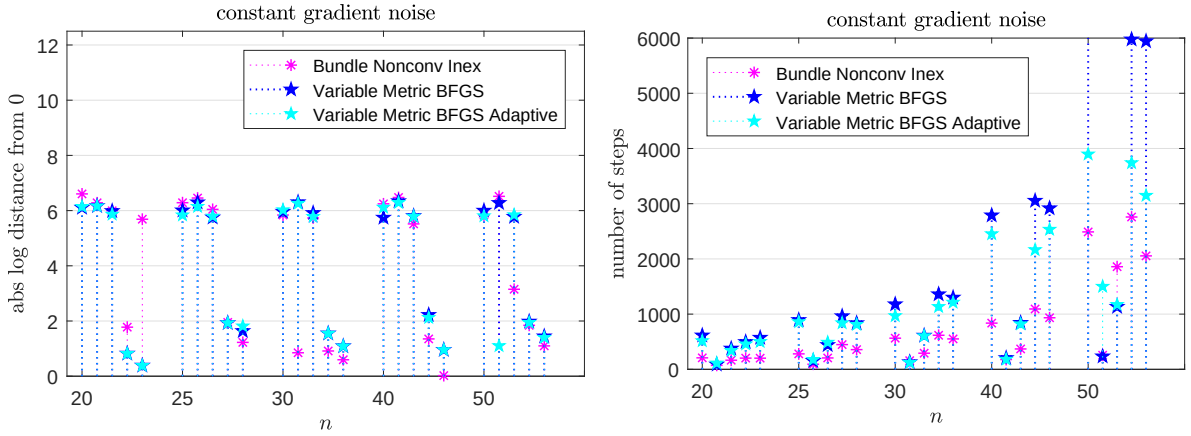

Figure 24: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant gradient noise*

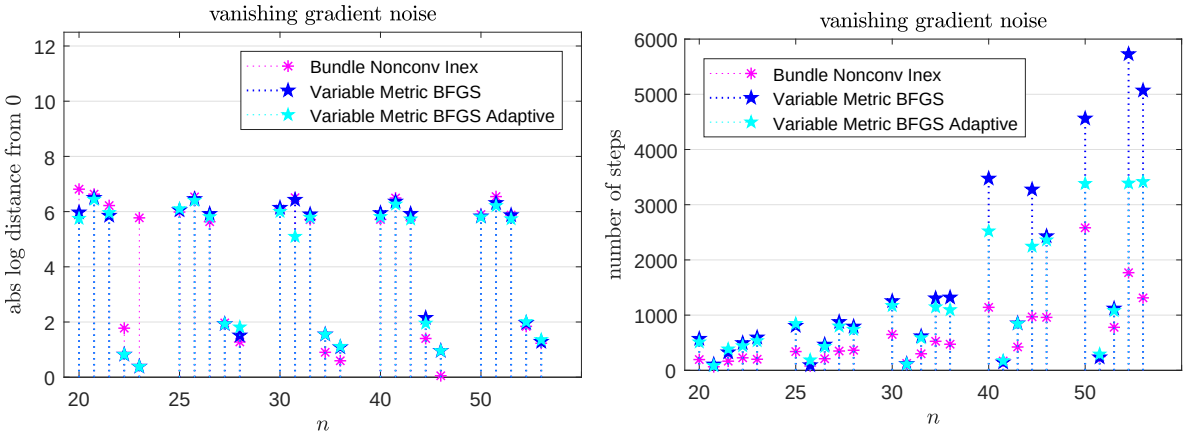

Figure 25: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing gradient noise*

References

[1] P. Apkarian, D. Noll, and O. Prot. A trust region spectral bundle method for nonconvex eigenvalue optimization. *SIAM J. Optim.*, 19(1):281–306, 2008.

[2] A. Bagirov, N. Karmitsa, and M. Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer International Publishing Cham, Switzerland, 2014.

[3] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992.

[4] F. Clarke. *Optimization and nonsmooth analysis*. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 1990.

[5] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Ann. Oper. Res.*, 153(1):235–256, 2007.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Camebridge, UK, 2000.

[7] W. de Oliveira and C. Sagastizábal. Bundle methods in the XXIst century: A bird's-eye view. *Pesc. Oper.*, 34(3):647–670, 2014.

[8] W. de Oliveira, C. Sagastizàbal, and C. Lemaréchal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Math. Program.*, 148:241–277, 2014.

[9] A. Fuduli, M. Gaudioso, and G. Giallombardo. A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optim. Method. Softw.*, 19(1):89–102, 2004.

[10] A. Fuduli, M. Gaudioso, and G. Giallombardo. Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *SIAM J. Optim*, 14(3):743–756, 2004.

[11] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer-Lehrbuch Masterclass. Springer, Berlin Heidelberg, Germany, 2002.

[12] S. Gunn. Support vector machines for classification and regression. Technical report, School of Electronics and Computer Science, University of Southampton, 1998.

[13] N. Haarala, K. Miettinen, and M. Mäkelä. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Math. Program.*, 109(1):181–205, 2007.

[14] W. Hare and C. Sagastizàbal. Computing proximal points of nonconvex functions. *Math. Program.*, 116:221–258, 2009.

[15] W. Hare and C. Sagastizàbal. A redistributed proximal bundle method for nonconvex optimization. *SIAM J. Optim.*, 20(5):2442–2473, 2010.

[16] W. Hare, C. Sagastizàbal, and M. Solodov. A proximal bundle method for nonsmooth nonconvex functions with inexact information. *Comput. Optim. Appl.*, 63:1–28, 2016.

[17] J. Heinonen. Lectures on Lipschitz analysis. Lectures at the 14th Jyväskylä Summer School in August 2004, 2004.

[18] M. Hintermüller. A proximal bundle method based on approximate subgradients. *Comput. Optim. Appl.*, 20:245–266, 2001.

[19] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin Heidelberg, Germany, 1993.

[20] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin Heidelberg, Germany, 2 edition, 1996.

[21] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, Camebridge, UK, 2012.

[22] A. Jofré, D. Luc, and M. Théra. ε-subdifferential and ε-monotonicity. *Nonlinear Anal.-Theor.*, 33(1):71–90, 1998.

[23] K. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Springer, Berlin Heidelberg, Germany, 1985.

[24] K. Kiwiel. An aggregate subgradient method for nonsmooth and nonconvex minimization. *J. Comput. Appl. Math.*, 14(3):391–400, 1986.

[25] K. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim*, 16(4):1007–1023, 2006.

[26] K. Kiwiel. Bundle methods for convex minimization with partially inexact oracles. Technical report, Systems Research Institute, Polish Academy of Sciences, 2010.

[27] K. Königsberger. *Analysis 1*. Springer, Berlin Heidelberg, Germany, 2003.

[28] G. Kunapuli. *A bilevel optimization approach to machine learning*. PhD thesis, Rensselaer Polytechnic Institute Troy, New York, NY, USA, 2008.

[29] C. Lemaréchal. Nonsmooth optimization and descent methods. IIASA research report, International Institute for Applied Systems Analysis, 1978.

[30] C. Lemaréchal and C. Sagastizábal. *An approach to variable metric bundle methods*, pages 144–162. Springer, Berlin Heidelberg, Germany, 1994.

[31] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: From conceptual to implementable forms. *Math. Program.*, 76(3):393–410, 1997.

[32] A. Lewis and M. Overton. Nonsmooth optimization via BFGS. *Submitted to SIAM Journal on Optimization*, 2008.

[33] A. Lewis and S. Wright. A proximal method for composite minimization. *Math. Program.*, 158(1-2):501–546, 2015.

[34] M. Lichman. UCI machine learning repository, 2013.

[35] J. Liesen and V. Mehrmann. *Linear Algebra*. Springer International Publishing, Cham, Switzerland, 2015.

[36] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *J. Optim. Theory Appl.*, 102(3):593–613, 1999.

[37] The MathWorks, Inc. *Documentation: Optimization Toolbox (quadprog)*, 2017.

[38] R. Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. In *Mathematical Programming Studies*, volume 17, pages 77–90. Springer, Berlin Heidelberg, Germany, 1982.

[39] R. Mifflin and C. Sagastizàbal. A science fiction story in nonsmooth optimization originating at IIASA. *Doc. Math.*, Extra Volume ISMP:291–300, 2012.

[40] G. Moore, C. Bergeron, and K. Bennett. Gradient-type methods for primal SVM model selection. Technical report, Rensselaer Polytechnic Institute, 2010.

[41] G. Moore, C. Bergeron, and K. Bennett. Model selection for primal SVM. *Mach. Learn.*, 85(1):175–208, 2011.

[42] Y. Nesterov and V. Shikhman. Algorithmic principle of least revenue for finding market equilibria. In *Optimization and Its Applications in Control and Data Sciences*, volume 115 of *Springer Optimization and Its Applications*, pages 381–435. Springer, Cham, Switzerland, 2016.

[43] J. Nocedal. Updating quasi-newton matrices with limited storage. *Math. Comput.*, 35(151):773–782, 1980.

[44] D. Noll. Cutting plane oracles to minimize non-smooth non-convex functions. *Set-Valued Var. Anal.*, 18(3-4):531–568, 2010.

[45] D. Noll. Bundle method for non-convex minimization with inexact subgradients and function values. In *Computational and Analytical Mathematics*, pages 555–592. Springer, New York, NY, USA, 2013.

[46] D. Noll and P. Apkarian. Spectral bundle method for non-convex maximum eigenvalue functions: first-order methods. *Math. Program.*, 104(2-3):701–727, 2005.

[47] D. Noll, O. Prot, and A. Rondepierre. A proximity control algorithm to minimize non-smooth and non-convex functions. *Pac. J. Optim.*, 4(3):571–604, 2012.

[48] J. Outrata, M. Kočvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*. Springer Science+Business Media, Dordrecht, The Netherlands, 1998.

[49] B. T. Polyak. *Introduction to Optimization*. Optimization Software , Inc., Publications Division, New York, NY, USA, 1987.

[50] R. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, USA, 1970.

[51] R. Rockafellar and R. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin Heidelberg, Germany, 3rd edition,

2009.

[52] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optim*, 2(1):121–152, 1992.

[53] A. J. Smola, S. V. N. Vishwanathan, and Q. V. Le. Bundle methods for machine learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1377–1384, Vancouver, British Columbia, Canada, 2007.

[54] M. Solodov. On approximations with finite precision in bundle methods for nonsmooth optimization. *J. Optim. Theory Appl.*, 119(1):151–165, 2003.

[55] J. Spingarn. Submonotone subdifferentials of Lipschitz functions. *T. Am. Math. Soc.*, 264:77 – 89, 1981.

[56] J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I*, volume 163 of *Die Grundlehren der mathematischen Wissenschaften*. Springer, Berlin Heidelberg, Germany, 1970.

[57] C. Teo, A. Smola, S. Vishwanathan, and Q. Le. Bundle methods for regularized risk minimization. *J. Mach. Learn. Res.*, 11:311–365, 2010.

[58] J. Treiman. Clarke's gradients and ε-subgradients in Banach spaces. *T. Am. Math. Soc.*, 294(1):65–65, 1986.

[59] M. Ulbrich and S. Ulbrich. *Nichtlineare Optimierung*. Birkhäuser Basel, Switzerland, 2012.

[60] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1998.

[61] V. Vapnik. An overview of statistical learning theory. *IEEE T. Neural. Networ.*, 10(5):988–999, 1999.

[62] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, NY, USA, 2013.

[63] J. Vlček and L. Lukšan. Globally convergent variable metric bundle method for nonconvex nondifferentiable unconstrained minimization. *J. Optim. Theory Appl.*, 111(2):407–430, 2001.

[64] G. Wachsmuth. On LICQ and the uniqueness of lagrange multipliers. *Oper. Res. Lett.*, 41(1):78 – 80, 2013.