
# Master thesis project report
# Bi-criterion $\ell_1/\ell_2$-norm optimization

**Joakim Jaldén**

**020923**

**IR-SB-EX-0221**

ROYAL INSTITUTE
OF TECHNOLOGY
Department of
Signals, Sensors & Systems
Signal Processing
S-100 44 STOCKHOLM

KUNGL TEKNISKA HÖGSKOLAN
Institutionen för
Signaler, Sensorer & System
Signalbehandling
100 44 STOCKHOLM

## Abstract

Many problems in engineering can be described as minimizing a norm of some sort. By far the most common of these problems are problems which require the minimization of a squared $\ell_2$-norm, *i.e.* the sum of squares. The choice of norm is sometimes rater arbitrary and the $\ell_2$-norm is often chosen due to its analytic simplicity, even though other choices of norm might yield more satisfactory results.

In this master thesis a particular class of optimization problems involving the $\ell_1$-norm as well as the squared $\ell_2$-norm is investigated. An efficient interior point algorithm based on the Mehrotra predictor corrector algorithm is developed for the solution of these. Furthermore several engineering applications from areas such as signal processing, statistics and control are presented in order to illustrate the usefulness of this work.

## Acknowledgements

This Master thesis work was carried out at Stanford University, CA, USA, as part of the electrical engineering program at the Royal Institute of Technology (KTH), Stockholm, Sweden. There are many people whose help has been essential to this thesis work, and also much appreciated. I would therefore like to thank...

Professor Stephen Boyd for being my supervisor and welcoming me into his research group at Stanford, for teaching me everything I know about convex optimization and without whose help and encouragement I could never have completed this thesis.

Professor Björn Ottersten at KTH for giving me the opportunity to travel to Stanford and carry out this Master thesis and also for allowing me to become a Ph.D. student in his group at S3, KTH, now that I reached the end of the Master thesis work.

Tomas Sköllermo for being my opponent, reading my report and giving me most useful feedback.

Jon Dattorro, Lin Xiao and Henrik Tidefelt for taking the time to read the report and giving me useful feedback and help. I especially wish to thank Jon without whose encouragement I would not have gotten involved in this field in the first place.

Haitham Hindi and Maryam Fazel Sarjoui for helpful discussions.

Anna Hammarwall for reading the final draft of the report and pointing out all my mistakes. This turned out be a greater task than I had expected.

Karin Demin and Denise Murphy for all your help. You have made my life so much simpler at so many occasions and for this I am truly grateful.

My parents for their encouragement and support. Also for helping me to, on short notice and on several occasions, move and store my entire apartment in their garage. I also like to thank my brother for keeping me updated on the status at home.

Joakim Jaldén
Stockholm, September 2002.

# Contents

# Chapter 1

# Introduction

The $\ell_2$-norm, *i.e.* the Euclidean norm or the sum of squares, has always played an important role in most fields of engineering. Many problems in engineering can be formulated as optimization problems involving the $\ell_2$-norm and these problems can be solved efficiently. The solution efficiency, rather than mathematical arguments, are in many applications the primary reason for choosing this norm.

Advances in convex programming and the ever decreasing cost of computation has however made large scale problems involving other types of norms computationally tractable.

The purpose of this master thesis is to investigate engineering problems that involve the use of the $\ell_1$-norm as well as the usual $\ell_2$-norm (see [CDS99, Fuc99, MM00, Tib96, Wil95]). This will be done by considering a convex optimization problem that is general enough to hold these engineering problems special cases.

The $\ell_1$-norm is the basis of many heuristics employed to obtain sparsity [BV01]. By this we mean that by properly incorporating the $\ell_1$-norm of a vector into our problem formulation we can make this vector contain few nonzero components. Applications involving the $\ell_1$-norm are generally designed to exploit this phenomena, directly or indirectly, since sparsity is often a desirable property. In areas such as statistics a sparse solution could tell us which factors are significant when one wish to predict a certain outcome [Tib96]. In signal processing a sparse solution might tell us that a signal can be effectively modeled by a relatively small set of basis functions [CDS99].

This report will examine and give example of application in areas such as signal processing, robust estimation, statistics and control. Some of these applications are extensively studied and algorithms for their solution have previously been developed. Using results from convex programming generalized algorithms can be created to obtain numerical solutions to this class of engineering problems. These generalized algorithms can be made to be competitive and in some cases superior to application specific algorithms [AR94, Dax91, LS96]. It is the purpose of this project is to investigate and present such general algorithms along with a proper presentation of the problems.

## 1.1 Overview

Chapter 2 defines the class of problems under consideration as a general convex bi-criterion problem involving an $\ell_1$ and an $\ell_2$ objective function. The tradeoff between these two objective function is discussed and single objective reformulations of the original problem are introduced.

In chapter 3 dual problems and conditions of optimality are derived for the single objective reformulations. Using these results some characteristics of the problems can be addressed in a straightforward manner. This will also provide the necessary foundation needed to construct algorithms for the numerical solution of these problems.

Chapter 4 will examine some of the characteristics of the solutions. Some of these characteristics are the primary reasons for using the $\ell_1$-norm in applications.

Algorithms used to obtain numerical solutions are developed in chapter 5. In particular a primal-dual interior-point method for the solution of two single objective reformulations of the bi-criterion problem is developed.

Chapter 6 deals with several applications that make use of the algorithms developed. Numerical results as well as brief discussions of the applications are given.

Finally some conclusions are made in chapter 7.

## 1.2 Notation

The notations used in the project report will be covered in this section. Fist the assumtion that all vectors are column vectors is made. That is

$$x \in \mathbf{R}^n$$

denotes the $n$ by 1 column vector $x$. The corresponding row vector is written as $x^T$. Furthermore $\mathbf{1}$ is used to denote a vector of all ones with dimension given by the context.

On a few occasions large systems of equations will be used. In order to simplify the notation

$$z = (x, y)$$

will be used to discuss $x$ and $y$ simultaneously. There might for instance be of interest to discuss a linear combination $z = z_1 + z_2$ of $z_1$ and $z_2$. Here it will be assumed to be understood that what is meant is $x = x_1 + x_2$ and $y = y_1 + y_2$.

The symbols $\succ$ and $\succeq$ are used when we discuss generalized inequalities are discussed. For vectors the generalized inequality will be the component wise inequality. That is for $x \in \mathbf{R}^n$ we will write $x \succ 0$ and $x \succeq 0$ and mean $x_i > 0$ and $x_i \geq 0$ for $i = 1, \ldots, n$ respectively. The extension $x \succeq y$ simply mean $x - y \succeq 0$.

In the case of matrices the same symbols imply positive definiteness and positive semi-definiteness, *i.e.* $P \succ 0$ and $Q \succeq 0$ means that P is positive definite and Q is positive semidefinite.

# Chapter 2

# Problem specification

In this chapter the class of problems under consideration is specified. The chosen approach is through the formulation of a convex bi-criterion problem.

Rigorous proofs of statements made will be carried out in chapter 3 and chapter 4. This chapter will however assume familiarity with basic concepts of convex optimization covered in [BV01, Roc70].

## 2.1 The bi-criterion problem

### 2.1.1 Objectives

A bi-criterion optimization problem is an optimization problem which has two competing objective functions $\varphi_1(x)$ and $\varphi_2(x)$. As stated earlier the focus is on one specific class of such problems involving an $\ell_1$-norm and an $\ell_2$-norm objective.

The first objective function, $\varphi_1(x)$, is

$$\varphi_1(x) = \|Cx - d\|_1 \tag{2.1}$$

where $\|z\|_1$ is defined by

$$\|z\|_1 = \sum_{i=0}^{p} |z_i|, \qquad z \in \mathbf{R}^p$$

The function $\varphi_1(x)$ can thus be described as the $\ell_1$-norm of an affine transformation of the optimization variable $x$.

The second objective, $\varphi_2(x)$, is

$$\varphi_2(x) = \frac{1}{2} \|Ax - b\|_2^2 \tag{2.2}$$

where $\|z\|_2^2$ is define by

$$\|z\|_2^2 = \sum_{i=0}^{m} z_i^2, \qquad z \in \mathbf{R}^m$$

The multiplication by a half in front of the $\ell_2$-norm in $\varphi_2(x)$ is not of theoretical importance but will simplify the formulas. Note that that $\varphi_2(x)$ can be described as the squared $\ell_2$-norm of and affine transformation of $x$.

The two objectives can also be viewed as a single vector-valued objective function $\varphi(x) = (\varphi_1(x), \varphi_2(x))$. Both forms will be used interchangeably throughout this report.

The matrices and vectors $A$, $b$, $C$ and $d$ are problem parameters of size

$$A \in \mathbf{R}^{m\times n}, \qquad b \in \mathbf{R}^m, \qquad C \in \mathbf{R}^{p\times n}, \qquad d \in \mathbf{R}^p. \tag{2.3}$$

The objective functions are convex functions of $x$. The objective is to minimize the objective functions over $\mathbf{R}^n$ or some affine subset $\mathcal{X}$ of $\mathbf{R}^n$. In the latter case the affine set is given by a matrix $F \in \mathbf{R}^{q\times n}$ and a vector $g \in \mathbf{R}^q$ such that

$$\mathcal{X} = \{x \mid Fx = g\}. \tag{2.4}$$

### 2.1.2 Pareto optimality

In a bi-criterion problem there is a certain ambiguity about what it means for an $x$ to be optimal. If there is a point $x$ such that neither $\varphi_1(x)$ or $\varphi_2(x)$ can be made smaller there is no question about the optimality of this point. This is however a rare event. More likely is that the two objectives will compete, that is in order to reduce $\varphi_1(x)$,$\varphi_2(x)$ must increase and vice verse. It is convenient to introduce the following definition.

Let $\mathcal{X}$ be a subset of $\mathbf{R}^n$ (with the possibility of $\mathcal{X} = \mathbf{R}^n$). Then $x^\star \in \mathcal{X}$ is said to be *Pareto optimal* on $\mathcal{X}$ if any $x \in \mathcal{X}$ such that $\varphi(x) \preceq \varphi(x^\star)$ implies $\varphi(x) = \varphi(x^\star)$ [SNT85]. Here $\varphi^\star = \varphi(x^\star)$ will be refered to as a *Pareto optimal value*.

The essence of Pareto optimality is that we can view a point $x^\star$ as being optimal in some sense if there are no other $x$ that makes both objectives lower.

### 2.1.3 The optimal tradeoff curve

An $x \in \mathcal{X}$ is called feasible and the set $\mathcal{O} \subseteq R_+^2$ defined by

$$\mathcal{O} = \{(\varphi_1, \varphi_2) \mid \varphi_1 = \varphi_1(x),\ \varphi_2 = \varphi_2(x),\ x \in \mathcal{X}\} \tag{2.5}$$

is called the set of achievable values. Thus for any Pareto optimal $x^\star$ the value $\varphi^\star = \varphi(x^\star)$ will be a *minimal element* of $\mathcal{O}$ meaning that there is no $\varphi \in \mathcal{O}$ such that $\varphi \preceq \varphi^\star$.

It is generally more convenient to consider a set $\mathcal{A}$ defined by

$$\mathcal{A} = \mathcal{O} + \mathbf{R}_+^2 = \left\{\varphi + t \in \mathbf{R}^2 \mid \varphi \in \mathcal{O},\ t \in \mathbf{R}_+^2\right\}.$$

$\mathcal{A}$ is a convex set with the same set of minimal elements as $\mathcal{O}$. The reason for considering this set is that $\mathcal{O}$ is generally not convex.

We will denote the set of minimal elements of $\mathcal{A}$ the optimal tradeoff curve denoted by $\Phi$. It can be shown [SNT85] that the optimal tradeoff curve is a closed connected set such that for every $\varphi_1$ there is at most one $\varphi_2$ such that $\varphi \in \Phi$, in other words that $\Phi$ is a curve in $\mathbf{R}^2$.

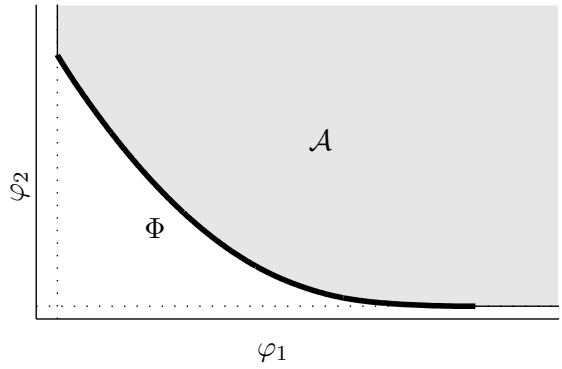


**Figure 2.1:** Optimal tradeoff curve, $\Phi$, and the set $\mathcal{A}$.

It can also be shown that if $\varphi_2$ is viewed as a function of $\varphi_1$ or vice verse it is a convex function. An example of this is given in figure 2.1 where the optimal tradeoff curve is shown.

## 2.2 Single objective reformulations

### 2.2.1 Weighted scalarization

There will be interested in obtaining Pareto optimal points $x^\star$. This can be accomplished in several different ways. One straight forward way is by weighted scalarization, that is for a parameter $\gamma \geq 0$ we look at the following problem:

$$\begin{array}{ll} \text{minimize} & \varphi_2(x) + \gamma\varphi_1(x) \\ \text{subject to} & x \in \mathcal{X}. \end{array} \tag{2.6}$$

Since both $\varphi_1(x)$ and $\varphi_2(x)$ are convex functions the objective of (2.6) is convex and since $\mathcal{X}$ is assumed to be an affine problem (2.6) is a convex optimization problem. The idea is that by choosing $\gamma \geq 0$ appropriately it will be able to obtain all Pareto optimal points as solutions to (2.6). The converse that any solution of (2.6) is a Pareto optimal point is true for $\gamma > 0$.

It is not obvious that the Pareto optimal point corresponding to the left edge of the curve can be obtained without having $\gamma$ equal to infinity. It is however later shown that this point can indeed be obtained for a finite $\gamma$.

Problem (2.6) will be refered to as the weighted scalarization reformulation of the bi-criterion problem or simply *the weighted scalarization problem.*

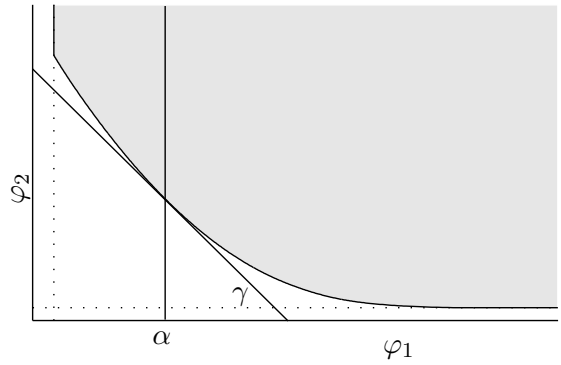


**Figure 2.2:** Graphical interpretation of weighed scalarization and constrained formulation.

### 2.2.2 Constrained formulation

Another way of obtaining Pareto optimal points is to minimize one of the objective functions, $\varphi_1(x)$, while imposing a constraint on the other, $\varphi_2(x)$. The problem can thus be formulated as

$$\begin{array}{ll} \text{minimize} & \varphi_2(x) \\ \text{subject to} & \varphi_1(x) \leq \alpha \\ & x \in \mathcal{X}. \end{array} \tag{2.7}$$

As with the previous formulation it can be stated that any Pareto optimal point can be obtained as the solution of (2.7). The criteria on $\alpha$ such that the converse is true is however slightly more complicated. Roughly speaking it will be have to required that the inequality constraint $\varphi_1(x) \leq \alpha$ is tight for the optimal point of (2.7).

Problem (2.7) will be refered to as the constrained reformulation of the bi-criterion problem or simply *the constrained problem.*

### 2.2.3 Graphical interpretations

There are graphical interpretations of the two procedures presented above. In the weighted scalarization case a Pareto optimal point $x^\star$ can be found by fixing the slope of the optimal tradeoff curve at $\varphi(x^\star)$. In the constrained case the Pareto optimal point is instead found by specifying the preferred value of $\varphi_1(x^\star)$. This is shown in figure 2.2.

By solving (2.6) an $\alpha$ can be obtained such that problem (2.7) yield the same solution $x^\star$. Later it will be shown that by solving (2.7) a $\gamma \geq 0$ will be found such that problem (2.6) for this $\gamma$ has the same solution. In other words fixing $\alpha$ an optimal value $\varphi^\star$ will be obtained as well as the slope of the tradeoff curve at $\varphi^\star$. In section 3.4 it is shown that this $\gamma$ is nothing more than the Lagrange dual variable corresponding to the constraint $\varphi_1(x) \leq \alpha$.

# Chapter 3

# Dual problems and conditions of optimality

In this chapter the dual problems and conditions of optimality of the weighted scalarization problem (2.6) and the constrained problem (2.7) are derrived. The chapter will also introduce equivalent quadratic programs (QP) to both of these. The rationale for this is that these QPs will be the basis for the algorithm developed in chapter 5. For this reason these QPs are chosen to simplify the development of the algorithms rater than to simplify the notation in this chapter. This may make the QPs appear unnecessarily large.

For both QPs first derive their dual problems are derrived as well as their Karush-Kuhn-Tucker (KKT) conditions of optimality. The duals of (2.6) and (2.7) will be derived from the duals of the QPs. This chapter assume working knowledge of the fundamental concepts of convex optimization such as duality and conditions of optimality. An introduction to these concepts can be found in [BV01].

## 3.1 Weighted scalarization

### 3.1.1 Problem reformulation

First look at problem (2.6) is treated, that is the convex optimization problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2+\gamma\left\|Cx-d\right\|_1 \\ \text{subject to} & Fx=g. \end{array} \tag{3.1}$$

In order to deal with this problem numerically an equivalent quadratic program with linear constraints of the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}u^Tu+\gamma\mathbf{1}^T(v+w) \\ \text{subject to} & u=Ax-b \\ & v-w=Cx-d \\ & Fx=g \\ & v\succeq 0 \\ & w\succeq 0 \end{array} \tag{3.2}$$

is formulated. Here the variables of optimization are $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$ and $v, w \in \mathbf{R}^p$.

It is easily verified that these two problems are equivalent in the sense that for any choice of $\gamma \geq 0$ an optimal point $x^\star$ of (3.2) will also be an optimal point of (3.1). The optimal value will also be the same in both cases. This implies that the dual problem of (3.2) will provide a tight lower bound on (3.1) as well as (3.2).

### 3.1.2 The dual problem

First the dual of problem (3.2) and (3.1) is derrived.

Let $\mathcal{L} = \mathcal{L}(x, u, v, w, \nu, \xi, \chi, \lambda, \nu)$ denote the Lagrangian function of problem 3.2. That is

$$\begin{aligned} \mathcal{L} = \quad & \tfrac{1}{2}u^T u + \gamma \mathbf{1}^T(v+w) + \nu^T(Ax - b - u) \\ & + \xi^T(Cx + d - v + w) + \chi^T(Fx - g) - \lambda^T v - \mu^T w \end{aligned}$$

for $\lambda, \mu \succeq 0$. Here the dual variables $\lambda, \mu, \xi, \in \mathbf{R}^p$, $\nu \in \mathbf{R}^m$ and $\chi \in \mathbf{R}^q$ are introduced.

By reordering the terms of $\mathcal{L}$ we get

$$\begin{aligned} \mathcal{L} = \quad & \tfrac{1}{2}u^T u - \nu^T u + (A^T\nu + C^T\xi + F^T\chi)^T x \\ & + (\gamma\mathbf{1} - \lambda - \xi)^T v + (\gamma\mathbf{1} - \mu + \xi)^T w - \nu^T b - \xi^T d - \chi^T g. \end{aligned} \tag{3.3}$$

In order to form the dual problem of (3.2) let $\mathcal{G} = \mathcal{G}(\nu, \xi, \chi, \lambda, \mu)$ be the Lagrange dual function defined by

$$\mathcal{G}(\nu, \xi, \chi, \lambda, \mu) = \inf_{x,u,v,w} \mathcal{L}(x, u, v, w, \nu, \xi, \chi, \lambda, \mu).$$

By using (3.3) it is seen that the Lagrangian function will be unbounded below in $(x, u, v, w)$ unless $A^T\nu + C^T\xi + F^T\chi = 0$, $\gamma\mathbf{1} - \lambda - \xi = 0$ and $\gamma\mathbf{1} - \mu + \xi = 0$. If these conditions are satisfied the expression for $\mathcal{G}$ simplifies to

$$\mathcal{G} = \inf_u \frac{1}{2}u^T u - \nu^T u - \nu^T b - \xi^T d - \chi^T g.$$

The above expression is minimized by $u = \nu$ which implies

$$\mathcal{G} = -\frac{1}{2}\nu^T\nu - b^T\nu - d^T\xi - g^T\chi.$$

The dual problem of (3.2) will therefore be

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\nu^T\nu - b^T\nu - d^T\xi - g^T\chi \\ \text{subject to} & A^T\nu + C^T\xi + F^T\chi = 0 \\ & \gamma\mathbf{1} - \lambda - \xi = 0 \\ & \gamma\mathbf{1} - \mu + \xi = 0 \\ & \lambda \succeq 0 \\ & \mu \succeq 0. \end{array} \tag{3.4}$$

This problem can be simplified by removing the variables $\lambda$ and $\mu$. The constraint on these variables becomes a set of constraint on $\xi$. The reformulated dual problem is

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\nu^T\nu - b^T\nu - d^T\xi - g^T\chi \\ \text{subject to} & A^T\nu + C^T\xi + F^T\chi = 0 \\ & \|\xi\|_\infty \leq \gamma. \end{array} \tag{3.5}$$

### 3.1.3 Optimality conditions

In this section conditions of optimality are derrived. By setting the derivatives of the Lagrangian function (3.3) with respect to $x, u, v$ and $w$ to zero the following conditions are obtained.

$$\begin{aligned}
\nabla_x \mathcal{L} &= A^T\nu + C^T\xi + F^T\chi = 0 \\
\nabla_u \mathcal{L} &= u - \nu = 0 \\
\nabla_v \mathcal{L} &= \gamma\mathbf{1} - \lambda - \xi = 0 \\
\nabla_w \mathcal{L} &= \gamma\mathbf{1} - \mu + \xi = 0.
\end{aligned}$$

These are the constraints that arose naturally in the derivation of the dual problem. By adding the constraints on the primal and dual variables to the complementary slackness conditions we get the KKT conditions for problem 3.2. These are

$$\begin{aligned}
A^T\nu + C^T\xi + F^T\chi &= 0 && \text{(3.6a)} \\
Ax - \nu - b &= 0 && \text{(3.6b)} \\
Cx - v + w - d &= 0 && \text{(3.6c)} \\
Fx - g &= 0 && \text{(3.6d)} \\
\lambda + \xi - \gamma\mathbf{1} &= 0 && \text{(3.6e)} \\
\mu - \xi - \gamma\mathbf{1} &= 0 && \text{(3.6f)} \\
\lambda_i v_i &= 0, \quad i = 1, \ldots, p && \text{(3.6g)} \\
\mu_i w_i &= 0, \quad i = 1, \ldots, p && \text{(3.6h)} \\
(\lambda, \mu, v, w) &\succeq 0 && \text{(3.6i)}
\end{aligned}$$

where the substitution $u = \nu$ has been made to eliminate $u$ from the system of equations.

Since problems (3.2) is a convex optimization problem with strictly feasible $(x, u, v, w)$ the KKT conditions are necessary and sufficient conditions of optimality. Thus any set of variables that satisfies (3.6) are primal-dual optimal with zero duality gap.

#### A compact form

From the optimality conditions of (3.2) and (3.4) the optimality conditions for problem (3.1) and (3.5) can be derrived. These are

$$\begin{aligned}
Fx - g &= 0 && \text{(3.7a)} \\
A^T\nu + C^T\xi + F^T\chi &= 0 && \text{(3.7b)} \\
\|\xi\|_\infty &\leq \gamma && \text{(3.7c)} \\
Ax - b &= \nu && \text{(3.7d)} \\
\xi^T(Cx - d) &= \gamma \|Cx - d\|_1 . && \text{(3.7e)}
\end{aligned}$$

Equations (3.7) are necessary and sufficient conditions for any primal-dual optimal point of (3.1) and (3.5)

Sufficiency is shown by first noting that any solution, $(x^\star, \nu^\star, \xi^\star, \chi^\star)$, of (3.7) satisfy the primal and dual constraints. Furthermore the duality gap of (3.1) and (3.5) can be written as

$$\frac{1}{2}\|Ax^\star - b\|_2^2 + \gamma\|Cx^\star - d\|_1 - (\frac{-1}{2}\nu^{\star T}\nu^\star - b^T\nu^\star - d^T\xi^\star - g^T\chi^\star).$$

By using equations (3.7d) and (3.7e) the above expression can be rewritten as

$$\frac{1}{2}\nu^{\star T}\nu^\star + \xi^{\star T}(Cx^\star - d) - (\frac{-1}{2}\nu^{\star T}\nu^\star - b^T\nu^\star - d^T\xi^\star - g^T\chi^\star)$$

which can be simplified to

$$(\nu^\star + b)^T\nu^\star + (Cx^\star)^T\xi^\star + g^T\chi^\star$$

and by using $\nu^\star + b = Ax^\star$ and $g = Fx^\star$ this can rewritten as

$$x^{\star T}(A^T\nu^\star + C^T\xi^\star + F^T\chi^\star)$$

which is zero due to (3.7b). Obtaining zero duality gap while being primal-dual feasible is sufficient for primal-dual optimality.

In order to show necessity it is first noted that for any primal-dual optimal point, $(x^\star, \nu^\star, \xi^\star, \chi^\star)$, of (3.1) and (3.5) it is trivial obtain a primal-dual optimal point of (3.2) and (3.4) by letting

$$\begin{array}{rcl} v_i^\star &=& \max((Cx^\star - d)_i, 0) \\ w_i^\star &=& \max(-(Cx^\star - d)_i, 0) \\ \lambda_i^\star &=& \gamma - \xi_i^\star \\ \mu_i^\star &=& \gamma + \xi_i^\star \end{array} \tag{3.8}$$

so all that is needed to conclude the proof is to show is that the constraints given by (3.7) follow from (3.6).

Equations (3.7a), (3.7b) and (3.7d) are given directly by (3.6) and equation (3.7c) follow from the positivity constraint on $\lambda$ and $\mu$. In order to show (3.7e) it is first shown that

$$\xi_i^\star(Cx^\star - d)_i = \gamma|(Cx^\star - d)_i|, \qquad i = 1, \ldots, p.$$

For $(Cx^\star - d)_i = 0$ this is obviously true. When $(Cx^\star - d)_i \neq 0$ it is first assumed that $(Cx^\star - d)_i > 0$ which implies $v_i^\star > 0$ and $w_i^\star = 0$. By using (3.6g) it is seen that $\lambda_i^\star = 0$ and therefore $\xi_i^\star = \gamma$. On the other hand if $(Cx^\star - d)_i < 0$ same argument leads to $w_i^\star > 0$, $\mu_i^\star = 0$ and $\xi_i^\star = -\gamma$ which proves the above statement.

## 3.2 The constrained problem

### 3.2.1 Problem reformulation

The analysis of the constrained problem (2.7) is similar to that of the weighted scalarization problems (2.6). The problem that should be solved is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax - b\right\|_2^2 \\ \text{subject to} & \left\|Cx - d\right\|_1 \leq \alpha \\ & Fx = g. \end{array} \tag{3.9}$$

As in the previous section the problem is first reformulated as a QP. A suitable form which closely resembles the previously used form is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}u^T u \\ \text{subject to} & \mathbf{1}^T(v+w) \leq \alpha \\ & u = Ax - b \\ & v - w = Cx - d \\ & Fx = g \\ & v \succeq 0 \\ & w \succeq 0. \end{array} \tag{3.10}$$

As for the weighted problem it is easily verified that the two problems yield the same optimal point $x^\star$ and optimal value.

### 3.2.2 The dual problem

Let $\mathcal{L} = \mathcal{L}(x, u, v, w, \nu, \xi, \chi, \lambda, \eta)$ denote the Lagrangian function of problem 3.10.

$$\begin{array}{rl} \mathcal{L} = & \frac{1}{2}u^T u + \eta(\mathbf{1}^T(v+w) - \alpha) + \nu^T(Ax + b - u) \\ & +\xi^T(Cx + d - v + w) + \chi^T(Fx - g) - \lambda^T v - \mu^T w \end{array}$$

for $\eta, \lambda, \mu \succeq 0$. Reordering the terms yields

$$\begin{array}{rl} \mathcal{L} = & \frac{1}{2}u^T u - \nu^T u + (A^T\nu + C^T\xi + F^T\chi)^T x \\ & +(\eta\mathbf{1} - \lambda - \xi)^T v + (\eta\mathbf{1} - \mu + \xi)^T w - \nu^T b - \xi^T d - \chi^T g - \eta\alpha. \end{array} \tag{3.11}$$

It is seen that the Lagrangian function is unbounded below in $(x, u, v, w)$ unless $A^T\nu + C^T\xi + F^T\chi = 0$, $\eta\mathbf{1} - \lambda - \xi = 0$ and $\eta\mathbf{1} - \mu + \xi = 0$. When these conditions are satisfied the Lagrange dual function $\mathcal{G} = \mathcal{G}(\nu, \xi, \chi, \lambda, \mu, \eta)$ is given by

$$\mathcal{G} = \inf_u \frac{1}{2}u^T u - \nu^T u - \nu^T b - \xi^T d - \chi^T g - \eta\alpha = -\frac{1}{2}\nu^T\nu - b^T\nu - d^T\xi - g^T\chi - \eta\alpha.$$

The dual problem can now be written as

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\nu^T\nu - b^T\nu - d^T\xi - g^T\chi - \eta\alpha \\ \text{subject to} & A^T\nu + C^T\xi + F^T\chi = 0 \\ & \eta\mathbf{1} - \lambda - \xi = 0 \\ & \eta\mathbf{1} - \mu + \xi = 0 \\ & \lambda \succeq 0 \\ & \mu \succeq 0 \\ & \eta \geq 0. \end{array} \tag{3.12}$$

As for the weighted bi-criterion problem we can simplify the problem by removing the variables $\lambda$ and $\mu$. 3.12 is then reduced to

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\nu^T\nu - b^T\nu - d^T\xi - g^T\chi - \eta\alpha \\ \text{subject to} & A^T\nu + C^T\xi + F^T\chi = 0 \\ & \|\xi\|_\infty \leq \eta. \end{array} \tag{3.13}$$

The constraint $\eta \geq 0$ is implicit in this form.

### 3.2.3 Optimality conditions

As before the Lagrangian is differentiated with respect to $x, u, v$ and $w$ to obtain the conditions

$$\begin{aligned}\nabla_x \mathcal{L} &= A^T\nu + C^T\xi + F^T\chi = 0\\ \nabla_u \mathcal{L} &= u - \nu = 0\\ \nabla_v \mathcal{L} &= \eta\mathbf{1} - \lambda - \xi = 0\\ \nabla_w \mathcal{L} &= \eta\mathbf{1} - \mu + \xi = 0.\end{aligned}$$

Adding the complementary slackness conditions and primal equality constraint the full KKT conditions of the problem are obtained as

$$\begin{aligned} A^T\nu + C^T\xi + F^T\chi &= 0 && && \text{(3.14a)}\\ Ax - \nu - b &= 0 && && \text{(3.14b)}\\ Cx - v + w - d &= 0 && && \text{(3.14c)}\\ Fx - g &= 0 && && \text{(3.14d)}\\ \lambda + \xi - \eta\mathbf{1} &= 0 && && \text{(3.14e)}\\ \mu - \xi - \eta\mathbf{1} &= 0 && && \text{(3.14f)}\\ \lambda_i v_i &= 0, && i = 1, \ldots, p && \text{(3.14g)}\\ \mu_i w_i &= 0, && i = 1, \ldots, p && \text{(3.14h)}\\ (\eta, \lambda, \mu, v, w) &\succeq 0 && && \text{(3.14i)}\\ \eta(\alpha - \mathbf{1}^T(v + w)) &= 0 && && \text{(3.14j)}\\ \alpha - \mathbf{1}^T(v + w) &\geq 0 && && \text{(3.14k)}\end{aligned}$$

where the substitution $u = \nu$ has been made.

The above conditions are necessary and sufficient for the primal-dual optimal points of (3.10) and (3.12). Note that this is true even in the case when $\alpha$ is chosen such that (3.10) is feasible but not strictly feasible. This follows from the fact that all inequality constraints are linear (Collary 28.2.2 of [Roc70]).

#### A compact form

As in the previous section the above conditions are used to obtain optimality conditions for problem (3.9) and (3.13). These are

$$\begin{aligned} Fx - g &= 0 && \text{(3.15a)}\\ A^T\nu + C^T\xi + F^T\chi &= 0 && \text{(3.15b)}\\ \|\xi\|_\infty &\leq \eta && \text{(3.15c)}\\ Ax - b &= \nu && \text{(3.15d)}\\ \xi^T(Cx - d) &= \eta\alpha. && \text{(3.15e)}\end{aligned}$$

Proving sufficiency of these conditions closely resembles the argument made in the previous section and is therefore omitted.

In proving necessity of these conditions it is noted that (3.15a), (3.15b), (3.15c) and (3.15d) are trivially obtained from (3.14).

For (3.15e) it is clear that $\eta^\star = 0$ imply $\xi^\star = 0$ for which the equality is satisfied. It can thus safely be assumed that $\eta^\star > 0$. By (3.14j) it is seen that $\alpha = \mathbf{1}^T(v^\star + w^\star) = \|Cx^\star - d\|_1$. Using the argument of the previous section it can be shown that

$$\xi_i^\star(Cx^\star - d)_i = \eta^\star|(Cx^\star - d)_i|$$

which implies

$$\xi^{\star T}(Cx^\star - d) = \eta^\star \|Cx^\star - d\|_1 = \eta^\star\alpha.$$

## 3.3 Interpretation of the dual variables

Equations (3.7) provide some interesting interpretations of the dual variables. Its first noted that by Hölder's inequality

$$\xi^T(Cx - d) \leq \|\xi\|_\infty \|Cx - d\|_1$$

for any $x \in \mathbf{R}^n$ and $\xi \in \mathbf{R}^p$. Equation (3.7c) and (3.7e) shows that for any primal-dual optimal point this inequality is tight. This implies that if $\xi^\star$ is any dual optimal $\xi$ then the function

$$f(x) = (\xi^\star)^T(Cx - d)$$

defines a supporting hyperplane to the epigraph of $\gamma \|Cx - d\|_1$ at any corresponding primal optimal point. Another aspect of this is that the dual optimal variable $\xi^\star$ is a subgradient of $\gamma \|y\|_1$ at $y^\star = Cx^\star - d$ with the property that any optimal point, $x^\star$, of the weighted scalarization problem (3.1) is also an optimal point of

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Ax - b\|_2^2 + (\xi^\star)^T(Cx - d) \\ \text{subject to} & Fx = g. \end{array} \tag{3.16}$$

The converse, that any solution of (3.16) is a solution of (3.1), is however not generally true.

## 3.4 Equivalence of the different formulations

Using (3.7) and (3.15) the statements made in the previous section about equivalence of the two single objective reformulations can be proven.

Assuming that the weighted scalarization problem (3.1) is solved in order to obtain a primal optimal $x$. Then this $x$ must satisfy (3.7) for some dual optimal $(\nu, \xi, \chi)$. Let $(x^\star, \nu^\star, \xi^\star, \chi^\star)$ be this primal-dual optimal point. If $\eta^star$ is chosen as $\eta^\star = \gamma$ and $\alpha = \|Cx^\star - d\|_1$ then $(x^\star, \nu^\star, \xi^\star, \eta^\star, \chi^\star)$ solves (3.15) This implies that $x^\star$ is also an optimal point of the constrained problem (3.9) for $\alpha = \|Cx^\star - d\|_1$.

Similarly if $x^\star$ is the optimal point of the constrained problem (3.9). This implies that there exists a dual optimal point, $(\nu^\star, \xi^\star, \eta^\star, \chi^\star)$, such that (3.15) hold. By choosing $\gamma = \eta^\star$ equation (3.7) will hold for $(x^\star, \nu^\star, \xi^\star, \chi^\star)$ which imply that $x^\star$ is an optimal point of the weighted scalarization problem (3.1) for $\gamma = \eta^\star$.

This shows that any Pareto optimal point can be obtained as the solution to either single objective reformulation. When the numerical algorithms are developed there has to be restritions on $\gamma$ and

$\alpha$ such that $\gamma$ is strictly positive and $\alpha$ is large enough such that problem (3.9) is strictly feasible. The reasons for this will be explained later. Even under these new constraints one is still able to obtain any Pareto optimal point arbitrarily close to these boundry points.

## 3.5 Summary

### 3.5.1 Weighted scalarization problem

The weighted scalarization problem formulation is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2+\gamma\left\|Cx-d\right\|_1 \\ \text{subject to} & Fx=g \end{array}$$

and has the corresponding dual problem

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\nu^T\nu-b^T\nu-d^T\xi-g^T\chi \\ \text{subject to} & A^T\nu+C^T\xi+F^T\chi=0 \\ & \left\|\xi\right\|_\infty\leq\gamma. \end{array}$$

An $x$ is primal optimal if and only if there exist $\nu$, $\xi$ and $\chi$ such that the following equations hold.

$$\begin{array}{rcl} Fx-g & = & 0 \\ A^T\nu+C^T\xi+F^T\chi & = & 0 \\ \left\|\xi\right\|_\infty & \leq & \gamma \\ Ax-b & = & \nu \\ \xi^T(Cx-d) & = & \gamma\left\|Cx-d\right\|_1. \end{array}$$

Similarly $(\nu,\xi,\chi)$ are dual optimal if and only if there exist an $x$ such that the above equations hold.

### 3.5.2 Constrained problem

The constrained problem formulation is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2 \\ \text{subject to} & \left\|Cx-d\right\|_1\leq\alpha \\ & Fx=g \end{array}$$

and has the corresponding dual problem

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\nu^T\nu-b^T\nu-d^T\xi-g^T\chi-\eta\alpha \\ \text{subject to} & A^T\nu+C^T\xi+F^T\chi=0 \\ & \left\|\xi\right\|_\infty\leq\eta. \end{array}$$

An $x$ is primal optimal if and only if there exist $\nu$, $\xi$, $\chi$ and $\eta$ such that the following equations hold.

$$\begin{array}{rcl} Fx-g & = & 0 \\ A^T\nu+C^T\xi+F^T\chi & = & 0 \\ \left\|\xi\right\|_\infty & \leq & \eta \\ Ax-b & = & \nu \\ \xi^T(Cx-d) & = & \eta\alpha. \end{array}$$

Similarly $(\nu,\xi,\chi,\eta)$ are dual optimal if and only if there exist an $x$ such that the above equations hold.

# Chapter 4

# Solution characteristics

In this chapter the characteristics of the problem and its solutions are discussed. Attempts will be made in order to give an intuitive understanding of these properties as well as solid mathematical proofs.

In chapter 3 equality between the two single objective reformulations was shown. Most proofs will only consider the weighted scalarization problem (2.6) since the solutions of this problem are also solutions to the constrained problem (2.7) for some choice of $\alpha$. In the last section these results will be extended to the constrained problem (2.7) and the relation between $\alpha$ and $\gamma$ will be established. The results are summarized at the end of the chapter.

## 4.1 Sparsity

It is well known that solutions to a problem of the form (2.6) tend to be sparse [AR94, Wil95]. This means that if $x^\star$ denotes the solution of (2.6) and

$$y^\star = Cx^\star - d$$

then $y^\star$ generally have many of its components exactly equal to zero. This, while being a rather vague statement, is a desirable property that most applications rely on, either explicitly or implicitly.

No quantitative proofs relating to this phenomena will be given. The reason for this is that it is hard to do so and that worst case upper bounds on the number of nonzero components of $y^\star$ does not tent to reflect the general case. Instead some intuition into why this happens will be given through simplified examples.

### 4.1.1 Some simple examples

Considering the function

$$f(x) = (x-1)^2 + \gamma|x|$$

defined on $x \in \mathbf{R}$. This is obviously an one-dimensional example of problem (2.6). Figure 4.1 shows

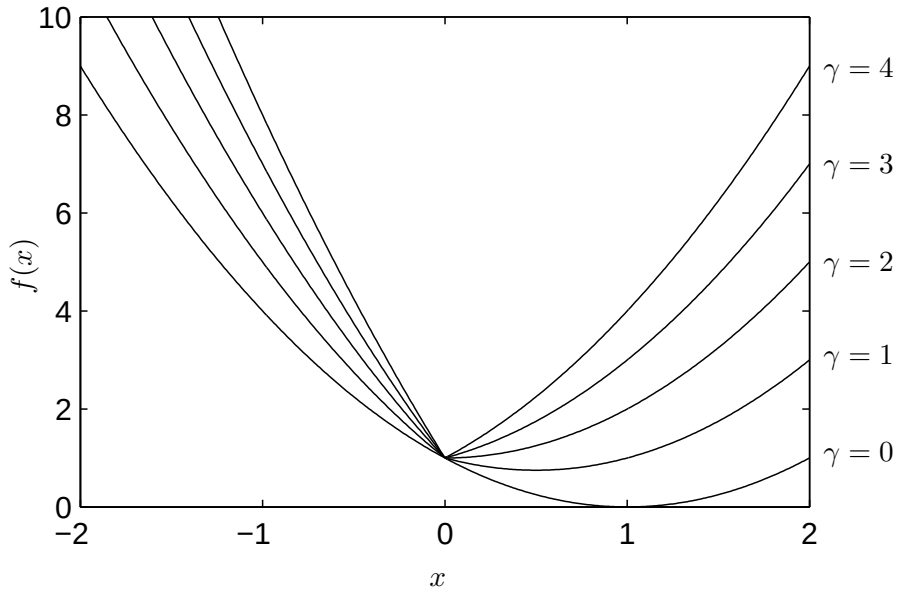


**Figure 4.1:** Function $f(x) = (x-1)^2 + \gamma|x|$ plotted for 5 different values of $\gamma$.

this function plotted for a set of different $\gamma$. Its seen that for every value of $\gamma \geq 2$ the minimum of $f(x)$ is achieved at $x^\star = 0$. For $\gamma \in [0, 2]$ the minimum, $x^\star$, is given by $x^\star = 1 - \frac{1}{2}\gamma$.

Another simple example is if $A \in \mathbf{R}^{n\times n}$ is chosen to be an orthogonal matrix, that is $A^T A = I$, $C = I$ and $d = 0$ for $C \in \mathbf{R}^{n\times n}$ and $d \in \mathbf{R}^n$. Under these conditions the weighted scalarization problem (2.6) has an explicit solution (see [Tib96]), $x^\star \in \mathbf{R}^n$, given by

$$x_i^\star = \text{sign}(x_i^0)(x_i^0 - \gamma)^+$$

where $x^0$ is the solution to $Ax = b$ given by $x^0 = A^T b$ and $(\cdot)^+$ is $\max(\cdot, 0)$. It is noted that as soon as $\gamma \geq x_i^0$ then the $i$'th component of $x^\star$ is fixed to zero. For any $\gamma \geq ||x^0||_\infty$ $x^\star = 0$.

The above examples show some important properties of the optimal points $x^\star$. First by increasing the factor $\gamma$ the number of zeros in $y^\star = C^\star + d$ is generally increased, *i.e.* this tends to increase the sparsity of $y^\star$. In the general case the number of nonzero elements of $y^\star$ is not however necessarily a monotonic function of $\gamma$.

Second, the solution $x^\star$, and thus $y^\star$, are a piecewise linear function of $\gamma$. This result turns out to be true in general provided that the solution $x^\star$ is unique. As could be expected the solution will also be piecewise linearly dependent on $\alpha$. Furthermore there will be a piecewise linear dependence between $\gamma$ and $\alpha$. This will be further discussed in section 4.6.

In both examples there is some $\gamma_{\max}$ for which the solution, $x^\star$, is not affected by changes in $\gamma$ as long as $\gamma \geq \gamma_{\max}$. This is also a general property of any solution of (2.6)

### 4.1.2 Sparsity as given by the dual problem

As noted earlier $\xi^\star$ of problem (3.5) acts as a sub-gradient of $\gamma \|y\|_1$ at $y^\star = Cx^\star - d$. This implies that whenever $\gamma > \xi_i^\star > -\gamma$ then $(Cx^\star - d)_i = 0$. Another way of obtaining this result is by

viewing the dual variables $\lambda_i^\star$ and $\mu_i^\star$ which must be strictly positive whenever $\xi_i^\star$ are in the interior of $[-\gamma, \gamma]$. This in turn imply that both $v_i^\star$ and $w_i^\star$ are zero due to the complementary slackness conditions.

The above implies a connection between the dual problem and the sparsity of the primal problem. The dual problem is given by a box-constrained quadratic program. Whenever the dual optimal variable $\xi^\star$ have a component, $\xi_i^\star$, not on the boundary of the box this will correspond to the primal problem having a zero in the $i$'th component of $Cx^\star - d$.

## 4.2 The optimal tradeoff curve revisited

Some statements made in chapter 2 will be used to obtain some of the proofs in this chapter Those statements will first be given some proper motivation.

A standard result from multiobjective optimization [SNT85] is that any Pareto optimal point can be obtained by weighted scalarization. More precisely, the problem

$$\begin{array}{ll} \text{minimize} & \delta\varphi_2(x) + \gamma\varphi_1(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

can be used to obtain every Pareto optimal point by varying $\delta \geq 0$ and $\gamma \geq 0$. This problem can be normalized such that $\delta = 1$ in all cases except when $\delta = 0$. This is however a non issue since if $\delta$ is fixed at 1 there is some finite $\gamma \geq 0$ for which we obtain the Pareto optimal point $x^\star$ for which $\varphi_1(x^\star)$ attains it minimum value over all $x \in \mathcal{X}$ is obtained. This means that the statement that any Pareto optimal point is obtainable as the solution of (2.6) for some $\gamma \geq 0$ can be made.

Conversely it is known that for any choice of $\delta > 0$ and $\gamma > 0$ the solution of the above problem is Pareto optimal. Here the problem can always be normalized so that $\delta = 1$.

One more property of the tradeoff curve that will be used later will be shown below. Let $x_1^\star$ and $x_2^\star$ be the solutions of the weighted scalarization problem (2.6) for $\gamma_1$ and $\gamma_2$ respectively. An intuitive statement is

$$\gamma_1 > \gamma_2 \quad \text{implies} \quad \varphi_1(x_2^\star) \geq \varphi_1(x_1^\star). \tag{4.1}$$

This is a well known result but since it is easily proven the proof will be carried out in full. The optimality of $x_1^\star$ and $x_2^\star$ implies

$$\begin{aligned} \varphi_2(x_2^\star) + \gamma_1\varphi_1(x_2^\star) &\geq \varphi_2(x_1^\star) + \gamma_1\varphi_1(x_1^\star) \\ \varphi_2(x_1^\star) + \gamma_2\varphi_1(x_1^\star) &\geq \varphi_2(x_2^\star) + \gamma_2\varphi_1(x_2^\star). \end{aligned}$$

Subtracting the right-hand side of second equation from the left-hand side of the first equation and vise verse leads to the equation

$$(\gamma_1 - \gamma_2)\varphi_1(x_2^\star) \geq (\gamma_1 - \gamma_2)\varphi_1(x_1^\star)$$

which proves the statement.

The geometric interpretation of this is that if a point is chosen on the tradeoff curve with a smaller slope than some other point this new point will be further to the right. In other words the optimal $\varphi_1$ as a function of $\gamma$ is non increasing.

## 4.3 Piecewise linearity

Due to its non-differentiability the $\ell_1$-norm is difficult to work with. Since there is interest in how the solutions are affected by variations in $\gamma \geq 0$ it will be shown that there is a finite partition of $\mathbf{R}_+$, $\{\Gamma_0, \ldots, \Gamma_N\}$, on which the analysis of the problem becomes greatly simplified for $\gamma \in \Gamma_i$, $i = 1, \ldots, N$.

To make the above statement more clear the optimality condition for the associated quadratic program (3.2) of the weighted scalarization problem are restated. These are

$$\begin{aligned} A^T\nu + C^T\xi + F^T\chi &= 0 \\ Ax - \nu &= b \\ Cx - v + w &= d \\ Fx &= g \\ \lambda + \xi &= \gamma\mathbf{1} \\ \mu - \xi &= \gamma\mathbf{1} \\ \lambda_i v_i &= 0, \qquad i = 1, \ldots, p \\ \mu_i w_i &= 0, \qquad i = 1, \ldots, p \\ (\lambda, \mu, v, w) &\succeq 0. \end{aligned} \tag{4.2}$$

Two index sets $J_{\lambda v}$ and $J_{\mu w}$ are defined as subset of $\{1, \ldots, p\}$ and the system

$$\begin{aligned} A^T\nu + C^T\xi + F^T\chi &= 0 \\ Ax - \nu &= b \\ Cx - v + w &= d \\ Fx &= g \\ \lambda + \xi &= \gamma\mathbf{1} \\ \mu - \xi &= \gamma\mathbf{1} \end{aligned} \tag{4.3}$$

with the constraints

$$\begin{aligned} \lambda_i = 0,\ v_i \geq 0, &\qquad i \in J_{\lambda v} \\ \lambda_i \geq 0,\ v_i = 0, &\qquad i \notin J_{\lambda v} \\ \mu_i = 0,\ w_i \geq 0, &\qquad i \in J_{\mu w} \\ \mu_i \geq 0,\ w_i = 0, &\qquad i \notin J_{\mu w} \end{aligned} \tag{4.4}$$

is considered. Clearly any solution of this system is also a solution of (4.2).

Assume that there is a closed convex subset, $\Gamma$, of $\mathbf{R}_+$ on which the solutions of (4.2) for all $\gamma \in \Gamma$ are solutions of (4.3) under the constraints (4.4) for some fixed $J_{\lambda v}$ and $J_{\mu w}$. Let $x^1$ be the solution of (3.2) for $\gamma^1 \in \Gamma$ and $x^2$ be the solution of (3.2) for $\gamma^2 \in \Gamma$ where $\gamma^1 \neq \gamma^2$. That is, there are $(\nu^k, \xi^k, \chi^k, v^k, w^k, \lambda^k, \mu^k)$ such that $x^k$ solves (4.2) for $\gamma^k$ where $k = 1, 2$. Let $x' = tx^1 + (1-t)x^2$, $\gamma' = t\gamma^1 + (1-t)\gamma^2$ and $(\nu', \xi', \chi', v', w', \lambda', \mu')$ be defined in the same way for some $t \in [0, 1]$. By linearity $x'$ solves (4.3) for $(\nu', \xi', \chi', v', w', \lambda', \mu')$ and $\gamma'$. Since the constraints given by (4.4) are convex $(x', \nu', \xi', \chi', v', w', \lambda', \mu')$ must satisfy these and $x'$ must therefore be an optimal $x$ of (3.2) for $\gamma'$.

This shows is that if $\mathcal{X}^\star(\gamma)$ is the set of optimal points of (2.6) for the given $\gamma$ then the set $\mathcal{Y}$ defined as

$$\mathcal{Y} = \{(\gamma, x) \mid \gamma \in \Gamma,\ x \in \mathcal{X}^\star(\gamma)\} \tag{4.5}$$

is a convex set. This has some interesting consequences. For instance if the solution, $x^\star$, of (2.6) is unique for $\gamma \in \Gamma$ then the solution $x^\star$ must be given by an affine function of $\gamma \in \Gamma$.

Another interesting and useful property is that since the sets $J_{\lambda v}$ and $J_{\mu w}$ are fixed for all $\gamma \in \Gamma$ the components of the vector $y^\star = Cx^\star - d$ will not change sign for $\gamma \in \Gamma$ meaning that if the function $f(\gamma)$ for $\gamma \in \Gamma$ is defined as

$$f(\gamma) = \|Cx^\star - d\|_1$$

where $x^\star \in \mathcal{X}^\star(\gamma)$, this function is an affine function from $\mathbf{R}$ to $\mathbf{R}$. It is not obvious at this point that this function is uniquely defined, *i.e.* that $\|Cx^\star - d\|_1$ is constant across $x^\star \in \mathcal{X}^\star(\gamma)$. This will be shown this in the next section.

### 4.3.1 Existence of the finite partition of $\mathbf{R}_+$

The results of this section will show that there always is a finite partition $\{\Gamma_0, \ldots, \Gamma_N\}$ of $\mathbf{R}_+$ such that the above assumptions hold for each $\Gamma_i$, $i = 1, \ldots, N$. The proof given is due to Mangasarian and Shiau and is essentially Lemma 3.1 of [MS87]. Since much of the analysis in this section is dependent on this the proof will be restated in the context of this thesis.

Fist the existence of this a finite partition will be shown. To be more precise, there exist a finite partition $\{\Gamma_0, \ldots, \Gamma_K\}$ of $[\gamma_l, \gamma_u]$ for any $0 \leq \gamma_l < \gamma_u$. Furthermore the existence of an upper bound on $K$ independent of $\gamma_l$ and $\gamma_u$ will be proven which in implies that there is a finite partition of $\mathbf{R}_+$.

Let $R(J_{\lambda v}, J_{\mu w})$ be the set of vectors, $r$, such that the system

$$\begin{bmatrix} A^T\nu + C^T\xi + F^T\chi \\ Ax - \nu \\ Cx - v + w \\ Fx \\ \lambda + \xi \\ \mu - \xi \end{bmatrix} = r \tag{4.6}$$

has a solution that satisfies the constraints given by (4.4). It is easily seen that $R(J_{\lambda v}, J_{\mu w})$ is a closed convex cone. It is also obvious that

$$\bigcup_{J_{\lambda v},\ J_{\mu w} \in \{1,\ldots,p\}} R(J_{\lambda v}, J_{\mu w})$$

is the set of all $r$ such that system (4.6) has a solution subject to the constraints

$$\begin{array}{rcll} \lambda_i v_i & = & 0 & i = 1 \ldots p \\ \mu_i w_i & = & 0 & i = 1 \ldots p \\ (\lambda, \mu, v, w) & \succeq & 0. & \end{array} \tag{4.7}$$

This is no more than stating that if the product of two terms are equal to zero then at least one of the terms must be equal to zero.

Let $r(\gamma)$ be given by

$$r(\gamma) = \begin{bmatrix} 0 \\ b \\ d \\ g \\ \gamma\mathbf{1} \\ \gamma\mathbf{1} \end{bmatrix} \tag{4.8}$$

and $\Gamma(J_{\lambda v}, J_{\mu w})$ be the set defined by

$$\Gamma(J_{\lambda v}, J_{\mu w}) = \{\gamma \mid \gamma \in [\gamma_l, \gamma_u],\ r(\gamma) \in R(J_{\lambda v}, J_{\mu w})\}.$$

Since $\Gamma(J_{\lambda v}, J_{\mu w})$ is the set of values of $\gamma$ that parameterize the intersection of a linesegment and the closed convex cone $R(J_{\lambda v}, J_{\mu w})$ the set $\Gamma(J_{\lambda v}, J_{\mu w})$ must be closed and convex. It might degenerate to a single point or the empty set.

Since (4.2) has a solution for all $\gamma \in [\gamma_l, \gamma_u]$ it follows that

$$[\gamma_l, \gamma_u] \subset \bigcup_{J_{\lambda v},\ J_{\mu w} \in \{1,\ldots,p\}} \Gamma(J_{\lambda v}, J_{\mu w}).$$

Let

$$L = \{[l_1, u_1], \ldots, [l_K, u_K]\}.$$

be the set of maximal intervals in $\{\Gamma(J_{\lambda v}, J_{\mu w}) \mid J_{\lambda v}, J_{\mu w} \in \{1, \ldots, p\}\}$. This means that a set $L$ is constructed from $\{\Gamma(J_{\lambda v}, J_{\mu w}) \mid J_{\lambda v}, J_{\mu w} \in \{1, \ldots, p\}\}$ by removing any element $[l_i, u_i]$ that is a subset of $[l_j, u_j]$ for some $j < i$, *i.e.* by removing any interval that is either contained in some other interval or simply the duplicate of a previous interval. Without loss of generality it can also be assumed that $l_{k-1} \leq l_k$ for all $k = 2, \ldots, K$. This is due to the fact that since if any $\gamma \in [\gamma_l, \gamma_u]$ belongs to some $\Gamma(J_{\lambda v}, J_{\mu w})$ it must lie inside some interval in $L$ due to the construction of $L$. Thus

$$[\gamma_l, \gamma_u] \subset \bigcup_{k=1}^{K} [l_k, u_k].$$

From the removal procedure performed to form $L$ it follows that $u_{k-1} < u_k$ since if this were false there would be some $[l_{k-1}, u_{k-1}] \in [l_k, u_k]$. This would contradict the fact that $[l_{k-1}, u_{k-1}]$ were a minimal element.

It can also be noted that $l_k \leq u_{k-1}$ since if this were not true $\gamma \in (u_{k-1}, l_k)$ would not be in any $\Gamma(J_{\lambda v}, J_{\mu w})$. This would be contradictory to the above statement. Using similar arguments it is also possible to show that $l_1 = \gamma_l$ and $u_K = \gamma_u$.

The above can be summarized by $l_1 = \gamma_l$, $l_{k-1} < l_k \leq u_{k-1} < u_k$ and $u_K = \gamma_u$. Let $\gamma_l = \gamma_0 < \ldots < \gamma_N = \gamma_u$ be the sorted numbers $l_1, u_1, \ldots, l_K, u_K$ with duplicates removed. These $\gamma$'s will provide a partition of $[\gamma_l, \gamma_u]$ with the desired properties since all $\gamma \in [\gamma_{k-1}, \gamma_k]$ will belong to the same $\Gamma(J_{\lambda v}, J_{\mu w})$.

$N$, and therefore $K$, has an upper bound independent of $\gamma_l$ and $\gamma_u$ since there are a finite number of possible sets $J_{\lambda v}$ and $J_{\mu w}$. This concludes the proof.

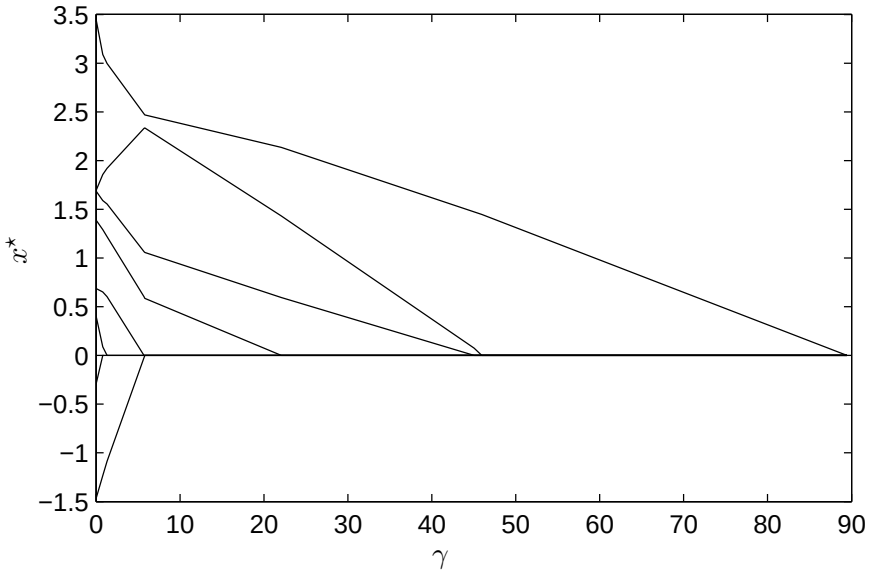


**Figure 4.2:** Example of piecewise linear solutions.

### 4.3.2 The solution as a function of $\gamma$

Some rather precise statements about the optimizing points, $x^\star$, of problem (2.6) can now be made.

If problem (2.6) yields a unique optimal point, $x^\star$, for every $\gamma \in [\gamma_l, \gamma_u]$ where $0 \leq \gamma_l < \gamma_u$, then $x^\star$, as a function of $\gamma$, must be a (Lipschitz) continuous piecewise linear function. The piecewise linearity is given by the previous proof and continuity is proven in [MS87]. The continuity follow from the fact that the sets $\Gamma_i$ are closed.

This is illustrated in figure 4.2 where the components of $x^\star$ are plotted as functions of $\gamma$ for an example problem in $\mathbf{R}^8$.

## 4.4 Uniqueness

In general the solution, $x^\star$, of (2.6) is not unique. It can however shown that for any $\gamma > 0$ the corresponding optimal value $\varphi^\star$ is unique. Geometrically this means that the tangent line given by $\gamma$ of the optimal tradeoff curve in figure (2.1) can touch the curve at only one point. In other words there can be no flat portions of the optimal tradeoff curve.

The above can be shown by an indirect proof. If $\gamma > 0$ if fixed there are Pareto optimal points $x_1^\star$ and $x_2^\star$ obtained by solving (2.6) for $\gamma$ such that $\varphi(x_1^\star) \neq \varphi(x_2^\star)$. This imply $\varphi_2(x_1^\star) \neq \varphi_2(x_2^\star)$ since if they were equal this would contradict the Pareto optimality of $x_1^\star$ or $x_2^\star$. Now consider the point $x(t) = tx_1^\star + (1-t)x_2^\star$ for some $t$ such that $0 < t < 1$.

Since $\|Cx^\star - d\|_1$ is given by an affine function for all $x \in \mathcal{X}^\star(\gamma)$ the following must hold.

$$\varphi_1(x(t)) = t\varphi_1(x_1^\star) + (1-t)\varphi_1(x_2^\star).$$

$\varphi_2(x(t))$ is a quadratic function of $t$ for $t \in \mathbf{R}$. That is, $\varphi_2(x(t)) = c_2t^2 + c_1t + c_0$ where $c_1 = 0$ if $c_2 = 0$. This is a consequence of the fact that $\varphi_2(x)$ is bounded below. Since $\varphi(x_1^\star) \neq \varphi(x_2^\star)$ it can be concluded that $c_2 \neq 0$ which imply that $\varphi_2(x(t))$ is strictly convex. Therefore

$$\varphi_2(x(t)) < t\varphi_2(x_1^\star) + (1-t)\varphi_2(x_2^\star)$$

for $0 < t < 1$.

Since by assumption, both $x_1^\star$ and $x_2^\star$ were optimal points which imply

$$\varphi_2(x_1^\star) + \gamma\varphi_1(x_1^\star) = \varphi_2(x_2^\star) + \gamma\varphi_1(x_2^\star)$$

but by the above statements

$$\begin{aligned} \varphi_2(x(t)) + \gamma\varphi_1(x(t)) &< t\varphi_2(x_1^\star) + (1-t)\varphi_2(x_2^\star) + t\gamma\varphi_1(x_1^\star) + (1-t)\gamma\varphi_1(x_2^\star) \\ &= \varphi_2(x_1^\star) + \gamma\varphi_1(x_1^\star) \end{aligned}$$

for $t$ such that $0 < t < 1$. This contradicts the optimality of $x_1^\star$ and $x_2^\star$.

Thus any given $\gamma > 0$ results in a unique optimal value $\varphi^\star$. This also proves the uniqueness of the function $f(\gamma)$ used in the previous section.

## 4.5 Existences of an upper bound on $\gamma$

This section will show that there is some $\gamma_{\max}$ such that the set of Pareto optimal points $\mathcal{X}^\star(\gamma)$ is the same for all $\gamma \geq \gamma_{\max}$.

By the proof given in section (4.3.1) it is already known that there is some $\gamma_{\max}$ such that there is an affine function, $f(\gamma)$, with the property that

$$f(\gamma) = \|Cx^\star - d\|_1$$

for $x^\star \in \mathcal{X}^\star(\gamma)$ whenever $\gamma \geq \gamma_{\max}$.

This function can not be decreasing, since it for some $\gamma$ would become negative which contradicts the non-negativity of the $\ell_1$-norm. It can neither be increasing since this would violate (4.1). Thus $f(\gamma)$ must be constant.

The fact that $f(\gamma)$ is constant imply that its value must be the optimal value of

$$\begin{array}{ll} \text{minimize} & \|Cx - d\|_1 \\ \text{subject to} & Fx = g \end{array} \tag{4.9}$$

over $x \in \mathbf{R}^n$. To see this note that $f(\gamma)$ is defined for all $\gamma \geq \gamma_{\max}$. If $f(\gamma)$ was not equal to the optimal value then there would be Pareto optimal points that were not obtainable for any $\gamma \geq 0$. Let $\mathcal{S}$ be the set of optimal points of (4.9). Then for $\gamma \geq \gamma_{\max}$ the set of optimal points of

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{subject to} & x \in \mathcal{S} \end{array} \tag{4.10}$$

is the same as the set of optimal points of (2.6). Since (4.10) is not dependent on $\gamma$ the set of optimal points of (2.6) must be the same for all $\gamma \geq \gamma_{\max}$

This existence result can also be obtained by applying Theorem 1 of ([MM79]) to a reformulation of the original problem. This was pointed out by Dax in ([Dax91]) for a simpler problem of the same form, but the proof can be extended to include our case.

In [OPT98] an explicit expression for this $\gamma_{\max}$ is given as $\gamma_{\max} = ||A^T b||_\infty$ in the case where $C = I$, $d = 0$ and in the absence of equality constraints. This explicit expression can be obtained from the dual problem (3.5) as the $\gamma$ for which the $\ell_\infty$ constraint becomes inactive. This corresponds to all components of the optimal $x^\star$ being equal to zero.

## 4.6 Relations between $\gamma$ and $\alpha$

### 4.6.1 Bounds on $\alpha$

It should be obvious that there is a lower bound, $\alpha_{\min}$, on $\alpha$ for which the constrained problem (2.7) becomes infeasible. This bound is always nonnegative due to the non-negativity of the $\ell_1$-norm. This alpha is obviously given by the objective value of

$$\begin{array}{ll} \text{minimize} & \|Cx - d\|_1 \\ \text{subject to} & Fx = g. \end{array}$$

Similarly there is an upper bound, $\alpha_{\max}$, on $\alpha$ for which the constraint becomes inactive if $\alpha \geq \alpha_{\max}$. Inactive in this case means that the objective value can not be made lower by removing the constraint. If $\mathcal{X}_{\text{ls}}^\star$ is the set of optimal points of

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{subject to} & Fx = g \end{array}$$

then this $\alpha_{\max}$ is given by the optimal value of

$$\begin{array}{ll} \text{minimize} & \|Cx - d\|_1 \\ \text{subject to} & x \in \mathcal{X}_{\text{ls}}^\star. \end{array}$$

This is illustrated in the context of the optimal tradeoff curve along with $\gamma_{\max}$ in figure 4.3.

### 4.6.2 $\gamma$ as a function of $\alpha$

As noted earlier any Pareto optimal point $x^\star$ obtained by solving the constrained problem (2.7) can also be obtained as the solution to the weighted scalarization problem (2.6) for the right choice of $\gamma$. In the next paragraph it is shown that the $\gamma$ chosen such that the weighted scalarization problem yield the same Pareto optimal value is given by a piecewise linear and nonincreasing function of $\alpha$. This implies that the solutions of (2.7) are piecewise linearly dependent on $\alpha$ similarly to the solutions of (2.6) being piecewise linearly dependent on $\gamma$.

It can be shown that if the $\ell_1$-constraint in (2.7) is active then $\eta^\star > 0$ and thus $\varphi_1(x^\star) = \alpha$. To obtain the same Pareto optimal value by weighted scalarization $\gamma$ should be chosen such that $\gamma = \eta^\star > 0$. This $\gamma$, as a function of $\alpha$, must be nonincreasing since if it was not it would violate (4.1). Also It can not be constant since this would violate the uniqueness of the Pareto

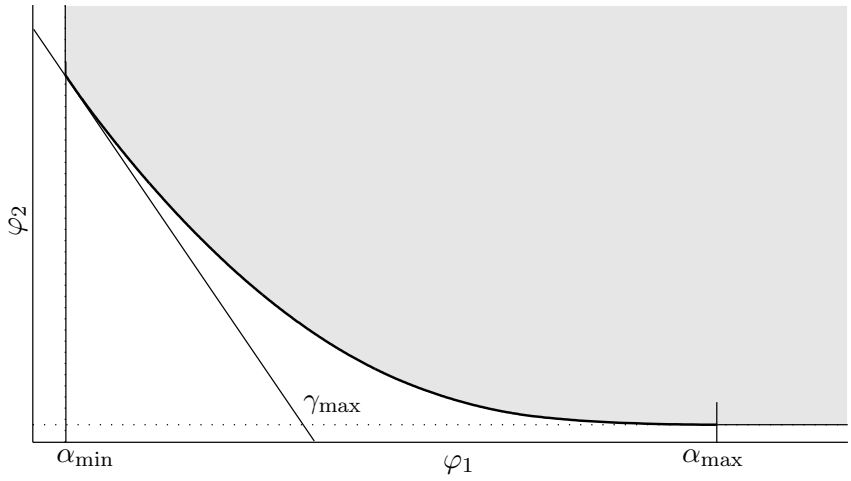


**Figure 4.3:** Graphical interpretation of $\gamma_{\max}$, $\alpha_{\min}$ and $\alpha_{\max}$.

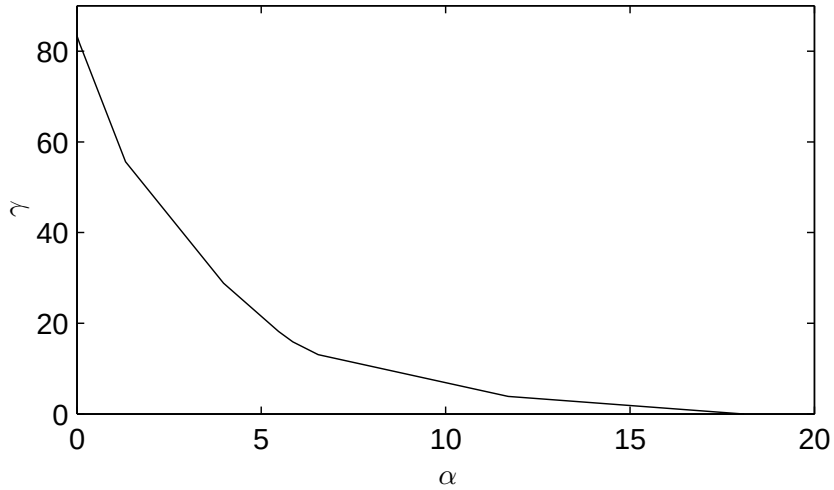


**Figure 4.4:** Relation between $\alpha$ and $\gamma$.

optimal values as a function of $\gamma$. Thus in the range $\alpha_{\min} \leq \alpha < \alpha_{\max}$ this function must be decreasing.

The above show that there is a one to one relationship between $\gamma$ and $\alpha$ in $\alpha_{\min} \leq \alpha < \alpha_{\max}$ and since we previously showed that $\alpha = \|Cx^\star - d\|_1 = f(\gamma)$ was piecewise linear this implies that $\gamma$ must be piecewise linearly dependent on $\alpha$ for $\alpha_{\min} \leq \alpha < \alpha_{\max}$.

If $\alpha \geq \alpha_{\max}$ then $\gamma = \eta^\star = 0$ which concludes the proof. The relationship between $\alpha$ and $\gamma$ is illustrated in figure 4.4.

## 4.7 Summary

The weighted scalarization problem is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2+\gamma\left\|Cx-d\right\|_1 \\ \text{subject to} & Fx=g. \end{array} \tag{4.11}$$

The following can be said about the above problem.

- Any solution $x^\star$ of (4.11) will tend to be sparse. That is many of the components of $x^\star$ will be equal to zero.
- If $x^\star$ is unique for all $\gamma \in [\gamma_l, \gamma_u]$ then $x^\star$ will be given by a continuous piecewise linear function of $\gamma$ on $[\gamma_l, \gamma_u]$.
- The values of $\frac{1}{2}\left\|Ax^\star-b\right\|_2^2$ and $\left\|Cx^\star-d\right\|_1$ are uniquely determined by $\gamma > 0$ even when the solution $x^\star$ is not unique.
- There exists a $\gamma_{\max}$ such that any solution $x^\star$ for $\gamma_1 \geq \gamma_{\max}$ is also a solution of (4.11) for $\gamma_2 \geq \gamma_{\max}$.

The constrained problem formulation is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2 \\ \text{subject to} & \left\|Cx-d\right\|_1 \leq \alpha \\ & Fx=g. \end{array} \tag{4.12}$$

The following can be said about this problem

- Any solution of (4.12) is also a solution of (4.11) for some $\gamma$. Furthermore this $\gamma$ is given by a nonincreasing, continuous and piecewise linear function of $\alpha \geq \alpha_{\min}$. Here $\alpha_{\min}$ is the smallest value of $\alpha$ for which (4.12) is feasible.

# Chapter 5

# Numerical solutions

In this chapter primal-dual interior point algorithms for the solution of the weighted scalarization and constrained problems are developed. These algorithms are based on the path following primal-dual algorithms described in the book *Primal-Dual Interior Point Methods* by Wright [Wri96] that in turn is based on the Mehrotra's predictor corrector algorithm [Meh92]. The notation is chosen to match the notation used in [Wri96] in the cases when it does not interfere with previously used notation.

This is not an attempt to introduce any new concepts in the area of quadratic programming but should instead be viewed as straight forward derivation of a interior-point algorithm as it applies to this particular problem. In fact a very similar algorithm is presented in [TVW01] for the solution of a generalized LASSO (see chapter 6) problem. To make the algorithms clearer only a few heuristics to accelerate the convergence of the algorithms will be presented. As shall be seen the basic interior-point framework will provide a very efficient algorithm.

## 5.1 Restrictions imposed on the original problem

Before describing the algorithms some restrictions to the original problem are made. The reasons for these restrictions will be made clear later in this chapter.

From this point on, in addition to assumptions made earlier, the following additional assumptions will also be made.

1. $\gamma$ is assumed to be strictly positive. If this is not true, *i.e.* $\gamma = 0$, the weighted scalarization problem would simplify to minimizing a quadratic function subject to linear equality constraints. This problem has an analytical solution that is readily computable.

2. $\alpha$ is assumed to be large enough to make the constrained problem feasible. A more elaborate discussion of this will be made later.

3. The matrices $A$ and $C$ will be assumed to have no common nullspace, that is

$$\mathcal{N}(A) \bigcap \mathcal{N}(C) = \{0\}.$$

The rationale of this will be explained later. For now it is sufficient to say that this relation will allow a set of linear equations to be solved in a more efficient manner.

## 5.2 Solving the weighted scalarization problem

### 5.2.1 The iterative process

The object of the interior-point algorithm is to produce a series of iterates

$$z^k = (x^k, \nu^k, \xi^k, \chi^k, v^k, w^k, \lambda^k, \mu^k)$$

that converges to a primal-dual optimal solution of problems (3.2) and (3.4). This is of course the same as the iterates converging to a solution of (3.6)

At each iteration a *search direction*, $\Delta z^k$, and a *step length*, $t^k$, will be computed and a new iterate, $z^{k+1}$, will be formed as

$$z^{k+1} = z^k + t^k \Delta z^k.$$

The algorithm will start with some initial value of $z^0$ and terminate once some final condition on $z^k$ is met.

In order to simplify notation the superscripts will be omitted from now on. It should be clear from the context what these superscripts should be.

The notation of $z$ as the collection of all variables are going to be used throughout this chapter. It is assumed that the definition of $z$ as

$$z = (x, \nu, \xi, \chi, v, w, \lambda, \mu)$$

will imply that the definitions of $\Delta z$ is

$$\Delta z = (\Delta x, \Delta\nu, \Delta\xi, \Delta\chi, \Delta v, \Delta w, \Delta\lambda, \Delta\mu).$$

Similar definitions will apply to sub- and superscripts.

### 5.2.2 The search direction

The search direction of the interior-point method has its origin in the steps obtained by applying Newton's method to solve a nonlinear system. The interest is in a nonlinear system of the form

$$\begin{aligned}
A^T\nu + C^T\xi + F^T\chi &= 0 && && \text{(5.1a)}\\
Ax - \nu - b &= 0 && && \text{(5.1b)}\\
Cx - v + w - d &= 0 && && \text{(5.1c)}\\
Fx - g &= 0 && && \text{(5.1d)}\\
\lambda + \xi - \gamma\mathbf{1} &= 0 && && \text{(5.1e)}\\
\mu - \xi - \gamma\mathbf{1} &= 0 && && \text{(5.1f)}\\
\lambda_i v_i &= \tau, && i = 1, \ldots, p && \text{(5.1g)}\\
\mu_i w_i &= \tau, && i = 1, \ldots, p && \text{(5.1h)}\\
(\lambda, \mu, v, w) &\succeq 0 && && \text{(5.1i)}
\end{aligned}$$

for some $\tau \geq 0$. For $\tau = 0$ these equations are the optimality conditions (3.6).

In order to simplify notation we, and following the conventions of interior-point programming, the following matrices are introduced.

$$\begin{aligned} V &= \operatorname{diag}(v_1, v_2, \ldots, v_p) \\ W &= \operatorname{diag}(w_1, w_2, \ldots, w_p) \\ \Lambda &= \operatorname{diag}(\lambda_1, \lambda_2, \ldots, v_p) \\ M &= \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_p). \end{aligned}$$

In general a capitalized letter where there is a lower case equivalent denotes the diagonal matrix obtained by taking the components of the lower case vector as diagonal elements.

The function $\mathcal{F}$ defined as

$$\mathcal{F}(z) = \begin{bmatrix} A^T\nu + C^T\xi + F^T\chi \\ Ax - \nu - b \\ Cx - v + w - d \\ Fx - g \\ \lambda + \xi - \gamma\mathbf{1} \\ \mu - \xi - \gamma\mathbf{1} \\ \Lambda V\mathbf{1} - \tau\mathbf{1} \\ MW\mathbf{1} - \tau\mathbf{1} \end{bmatrix}$$

enables system (5.1) to be written in a compact form as

$$\begin{aligned} \mathcal{F}(z) &= 0 \\ (\lambda, \mu, v, w) &\succeq 0. \end{aligned}$$

The search direction, $\Delta z$, is defined as the solution of

$$\mathcal{J}(z)\Delta z = -\mathcal{F}(z)$$

where $\mathcal{J}$ denotes the Jacobian matrix of $\mathcal{F}$. This is the equation that determines the steps in Newton's method for finding roots of nonlinear systems.

In its full form the above system is

$$\begin{bmatrix} 0 & A^T & C^T & F^T & 0 & 0 & 0 & 0 \\ A & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & -I & I & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & I & 0 \\ 0 & 0 & -I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & \Lambda & 0 & V & 0 \\ 0 & 0 & 0 & 0 & 0 & M & 0 & W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\nu \\ \Delta\xi \\ \Delta\chi \\ \Delta v \\ \Delta w \\ \Delta\lambda \\ \Delta\mu \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_2 \\ -r_1 \\ -r_p \\ -r_{\xi\lambda} \\ -r_{\xi\mu} \\ -r_{\lambda v} \\ -r_{\nu w} \end{bmatrix} \tag{5.2}$$

where

$$r_d = A^T\nu + C^T\xi + F^T\chi \tag{5.3a}$$

$$r_2 = Ax - \nu - b \tag{5.3b}$$

$$r_1 = Cx - v + w - d \tag{5.3c}$$

$$
\begin{aligned}
r_p &= Fx - g && \text{(5.3d)}\\
r_{\xi\lambda} &= \lambda + \xi - \gamma\mathbf{1} && \text{(5.3e)}\\
r_{\xi\mu} &= \mu - \xi - \gamma\mathbf{1} && \text{(5.3f)}\\
r_{\lambda v} &= \Lambda V\mathbf{1} - \tau\mathbf{1} && \text{(5.3g)}\\
r_{\nu w} &= MW\mathbf{1} - \tau\mathbf{1}. && \text{(5.3h)}
\end{aligned}
$$

The left hand side vectors are called the *residuals*. The object is thus to make these residuals small, preferably zero, subject to the constraints on the variables.

Later in this chapter it will be shown how to efficiently solve system (5.2) in order to obtain the search direction.

### 5.2.3 The central path

It is not immediately clear what function the parameter $\tau$ has in system (5.1). The intuitive thing to do would be to have $\tau = 0$ since the solution of (5.1) would then produce a primal-dual optimal solution. The function of $\tau$ becomes clear when one considers the constraints

$$(\lambda, \mu, v, w) \succeq 0.$$

In order to converge to the right solution, *i.e.* the one that satisfy these constraints, all iterates must satisfy the above constrains. This will impose restrictions on the step length, $t$, since a step equal to $\Delta z$, *i.e.* $t = 1$, would generally place the next iterate outside the set defined by these constrains.

Equation (5.1g) and (5.1h) for $\tau = 0$ places the solution on the boundary of the primal-dual feasible set. Due to this the last iterates must lie close to the boundary of this set and force the step length, $t$, to be very small not to violate any constraints. This would in fact be most likely to cause the algorithm not to converge at all.

On the other hand a value of $\tau > 0$ will place the solution of (5.1) strictly inside the feasible set. The fundamental idea of the interior-point algorithm is to take steps towards the solution of (5.1) while bringing $\tau$ to zero in a controlled fashion and thus be able to chose $t$ large at each iteration. In the language of interior-point programming $\tau > 0$ defines through the solution of (5.1) a set of points in the interior of the feasible set called the *central path*. Roughly speaking the central path parameterized by $\tau > 0$ guides the iterates to the solution of the optimality conditions as $\tau$ goes to zero.

### 5.2.4 The affine scaling direction and the centering parameter

The choice of $\tau$ is yet to be specified. It turns out that a good choice of $\tau$ is

$$\tau = \sigma\rho$$

where

$$\rho = \frac{\lambda^T v + \mu^T w}{2p}$$

and $\sigma \in [0,1]$. $\sigma$ is called the *centering parameter* for reasons that will soon become clear. $\rho$ is the average value of the products $\lambda_i v_i$ and $\mu_i w_i$ and by choosing $\sigma = 1$ the step taken will move all products closer to their average value. The average value itself will not decrease. This is called centering, since it will affectively move the iterates closer to the central path.

On the other hand by choosing $\sigma$ small $\rho$ will be decreased for the next iteration. This is most desirable since a small value of $\rho$ implies that the products $\lambda_i v_i$ and $\mu_i w_i$ are small which in imply that the iterate is close to the primal-dual optimum.

There is no satisfactory theory on how to optimally chose $\sigma$ to obtain a good balance between centering and reduction of the products. Bad centering will force the step length $t$ of each step to be small in order not to violate (5.1i) while excessive centering will have the effect that $\rho$ is brought toward 0 slower than necessary. A simple but effective heuristic to adaptively choose $\sigma$ at each step will be used.

The choice of $\sigma$ is made by first looking at the effect of setting $\tau = 0$. The direction obtained by doing so is called the *affine scaling direction*. That is, solve equation 5.2 using $\tau = 0$ to obtain $\Delta z_{\text{aff}}$. Then choose a step length $t_{\text{aff}}$ as

$$t_{\text{aff}} = \max\{t \in [0,1] \mid (v, w, \lambda, \mu) + t(\Delta v_{\text{aff}}, \Delta w_{\text{aff}}, \Delta\lambda_{\text{aff}}, \Delta\mu_{\text{aff}}) \succeq 0\}.$$

This is the largest step less than 1 that can be taken in the affine scaling direction without violating (5.1i). Let $z_{\text{aff}}$ be

$$z_{\text{aff}} = z + t_{\text{aff}}\Delta z_{\text{aff}}.$$

$z_{\text{aff}}$ would be the next iterate if the affine scaling direction were chosen as search direction and the step length was maximized.

As a measure of the effectiveness of the affine scaling direction $\rho_{\text{aff}}$ is defined as

$$\rho_{\text{aff}} = \frac{\lambda_{\text{aff}}^T v_{\text{aff}} + \mu_{\text{aff}}^T w_{\text{aff}}}{2p}.$$

If $\rho_{\text{aff}} \ll \rho$ a step close to the affine scaling direction would significantly reduce $\rho$ and $\sigma$ should be small. If on the other hand $\rho_{\text{aff}} \approx \rho$ there would be little benefit in moving in the affine scaling direction, most likely due to bad centering, and $\sigma$ should be large in order to center the components and enable a larger decrease at the next iteration.

The heuristic for choosing $\sigma$ proposed by Mehrotra [Meh92] which has proven itself through extensive testing is

$$\sigma = \left(\frac{\rho_{\text{aff}}}{\rho}\right)^3. \tag{5.4}$$

### 5.2.5 The corrector and centering steps

$\tau$ should now be chosen as $\tau = \sigma\rho$ and (5.2) be solved to obtain the search direction. The residuals $r_{\lambda v}$ and $r_{\mu w}$ defined by (5.3g) and (5.3g) would then become:

$$\begin{aligned} r_{\lambda v} &= \Lambda V\mathbf{1} - \sigma\rho\mathbf{1} \\ r_{\nu w} &= MW\mathbf{1} - \sigma\rho\mathbf{1}. \end{aligned}$$

There is however a trick that will speed up the convergence of the interior point algorithm quite noticeably. This is by observing how the products $\lambda_i v_i$ and $\mu_i w_i$ are affected by a full affine scaling step, that is

$$\begin{aligned}(\lambda_i + \Delta\lambda_i^{\text{aff}})(v_i + \Delta v_i^{\text{aff}}) &= \lambda_i v_i + \Delta\lambda_i^{\text{aff}} v_i + \Delta v_i^{\text{aff}}\lambda_i + \Delta\lambda_i^{\text{aff}}\Delta v_i^{\text{aff}} = \Delta\lambda_i^{\text{aff}}\Delta v_i^{\text{aff}} \\ (\mu_i + \Delta\mu_i^{\text{aff}})(w_i + \Delta w_i^{\text{aff}}) &= \mu_i w_i + \Delta\mu_i^{\text{aff}} w_i + \Delta w_i^{\text{aff}}\mu_i + \Delta\mu_i^{\text{aff}}\Delta w_i^{\text{aff}} = \Delta\mu_i^{\text{aff}}\Delta w_i^{\text{aff}}\end{aligned}$$

where $\Delta\lambda_i^{\text{aff}} v_i + \Delta v_i^{\text{aff}}\lambda_i = -\lambda_i v_i$ and $\Delta\mu_i^{\text{aff}} w_i + \Delta w_i^{\text{aff}}\mu_i = -\mu_i w_i$ were used. Since all other residuals correspond to linear equations in (5.3) these would become equal to zero if a full affine scaling step was made. Redefining $r_{\lambda v}$ and $r_{\mu w}$ to be

$$r_{\lambda v} = \Lambda V\mathbf{1} + \Delta\Lambda^{\text{aff}}\Delta V^{\text{aff}}\mathbf{1} - \sigma\rho\mathbf{1} \tag{5.5a}$$

$$r_{\nu w} = MW\mathbf{1} + \Delta M^{\text{aff}}\Delta W^{\text{aff}}\mathbf{1} - \sigma\rho\mathbf{1} \tag{5.5b}$$

will generally make the pairwise products closer to zero than what the original definitions of $r_{\lambda v}$ and $r_{\mu w}$ would have accomplished. The terms added are part of what is called the *corrector step*. A proper motivation of this is given in [Wri96].

Solving (5.2) using the new definitions of $r_{\lambda v}$ and $r_{\mu w}$ results in the final search direction $\Delta z$. When the search direction is calculated the step length $t$ is chosen as

$$t = \max\{t \in [0,1] \mid (v, w, \lambda, \mu) + t(\Delta v, \Delta w, \Delta\lambda, \Delta\mu) \succeq \epsilon(v, w, \lambda, \mu)\} \tag{5.6}$$

for some $\epsilon > 0$. The $\epsilon$ is simply there to prevent the iterate from ending up on the boundary of the set defined by (5.1i). In the language of interior-point programming the iterates will be required to lie within a $\mathcal{N}_{-\infty}$ neighborhood of the central path. $\epsilon$ is typically of size 0.01 but testing shows that for the problems at hand the algorithm converges faster for more aggressive values and a value of $\epsilon = 0.001$ is chosen.

The iterates are then updated as

$$z^{k+1} = z^k + t\Delta z^k.$$

### 5.2.6 Initial conditions and termination criterion

In order to start the algorithm a suitable starting point that strictly satisfies (5.1i) must be obtained. The chosen starting point is

$$\begin{aligned} x^0 = 0, \quad \nu^0 = 0, \quad \xi^0 = 0, \quad \chi^0 = 0, \\ v^0 = \mathbf{1}, \quad w^0 = \mathbf{1}, \quad \lambda^0 = \gamma\mathbf{1}, \quad \nu^0 = \gamma\mathbf{1}. \end{aligned} \tag{5.7}$$

This starting point will make the residuals $r_d$, $r_{\xi\lambda}$ and $r_{\xi\mu}$ all equal to 0. Since they all correspond to linear equations they will remain 0 for all iterates.

The algorithm will terminate once

$$\frac{\|Fx - g\|_2}{1 + \|g\|_2} < \epsilon \tag{5.8a}$$

$$A^T\nu + C^T\xi + F^T\chi = 0 \tag{5.8b}$$

$$\|\xi\|_\infty \leq \gamma \tag{5.8c}$$

$$\frac{\|Ax-\nu-b\|_2}{1+\|b\|_2} < \epsilon \tag{5.8d}$$

$$\frac{\gamma\|Cx-d\|_1-\xi^T(Cx-d)}{1+\gamma\|Cx-d\|_1} < \epsilon. \tag{5.8e}$$

That is, the algorithm will terminate once the optimality conditions given by (3.7) are satisfied up to some relative tolerance $\epsilon > 0$. Since (5.8b) and (5.8c) are satisfied at each iteration we do not explicitly have to check this condition for termination.

### 5.2.7 Computing the search direction

This section will show how to solve the equations needed to obtain a search direction. Although large, system (5.2) is highly sparse and can be solved efficiently. The first step is to rewrite (5.2) as a system of equations.

$$A^T\Delta\nu + C^T\Delta\xi + F^T\Delta\chi = -r_d \tag{5.9a}$$
$$A\Delta x - \Delta\nu = -r_2 \tag{5.9b}$$
$$C\Delta x - \Delta v + \Delta w = -r_1 \tag{5.9c}$$
$$F\Delta x = -r_p \tag{5.9d}$$
$$\Delta\lambda + \Delta\xi = -r_{\xi\lambda} \tag{5.9e}$$
$$\Delta\mu - \Delta\xi = -r_{\xi\mu} \tag{5.9f}$$
$$\Lambda\Delta v + V\Delta\lambda = -r_{\lambda v} \tag{5.9g}$$
$$M\Delta w + W\Delta\mu = -r_{\nu w}. \tag{5.9h}$$

Equation (5.9e) and (5.9f) can be rewritten as

$$\begin{aligned}\Delta\lambda &= -\Delta\xi - r_{\xi\lambda}\\ \Delta\mu &= \Delta\xi - r_{\xi\mu}.\end{aligned}$$

This is inserted into (5.9g) and (5.9h) to obtain

$$\begin{aligned}\Lambda\Delta v - V\Delta\xi - Vr_{\xi\lambda} &= -r_{\lambda v}\\ M\Delta w + W\Delta\xi - Wr_{\xi\mu} &= -r_{\nu w}.\end{aligned}$$

Since $\Lambda$ and $M$ are positive definite diagonal matrices they are easily inverted and

$$\begin{aligned}\Delta v &= \Lambda^{-1}V\Delta\xi + \Lambda^{-1}Vr_{\xi\lambda} - \Lambda^{-1}r_{\lambda v}\\ \Delta w &= -M^{-1}W\Delta\xi + M^{-1}Wr_{\xi\mu} - M^{-1}r_{\nu w}.\end{aligned}$$

This is inserted into (5.9c) to obtain

$$C\Delta x - (\Lambda^{-1}V\Delta\xi + \Lambda^{-1}Vr_{\xi\lambda} - \Lambda^{-1}r_{\lambda v}) + (-M^{-1}W\Delta\xi + M^{-1}Wr_{\xi\mu} - M^{-1}r_{\nu w}) = -r_1$$

which simplifies to

$$C\Delta x - (\Lambda^{-1}V + M^{-1}W)\Delta\xi = -(r_1 + \Lambda^{-1}r_{\lambda v} - M^{-1}r_{\nu w} - \Lambda^{-1}Vr_{\xi\lambda} + M^{-1}Wr_{\xi\mu}).$$

Introducing the notation

$$\begin{aligned}D &= \Lambda^{-1}V + M^{-1}W\\ r_x &= r_1 + \Lambda^{-1}r_{\lambda v} - M^{-1}r_{\nu w} - \Lambda^{-1}Vr_{\xi\lambda} + M^{-1}Wr_{\xi\mu}\end{aligned}$$

and noting that $D$ is positive definite allows $\Delta\xi$ to be written as

$$\Delta\xi \;=\; D^{-1}C\Delta x + D^{-1}r_x.$$

From equation (5.9b) the expression for $\nu$ is obtained as

$$\Delta\nu \;=\; A\Delta x + r_2$$

Using the expressions for $\Delta\nu$ and $\Delta\xi$ equation (5.9a) becomes

$$A^T(A\Delta x + r_2) + C^T(D^{-1}C\Delta x + D^{-1}r_x) + F^T\chi \;=\; -r_p$$

which simplifies to

$$(A^TA + C^TD^{-1}C)\Delta x + F^T\Delta\chi \;=\; -A^Tr_2 - C^TD^{-1}r_x - r_p.$$

Introducing the notation

$$\begin{aligned} H &= A^TA + C^TD^{-1}C \\ h &= A^Tr_2 + C^TD^{-1}r_x + r_p \end{aligned}$$

equations (5.9a) and (5.9d) can be written as

$$\begin{bmatrix} H & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\chi \end{bmatrix} = \begin{bmatrix} -h \\ -r_p \end{bmatrix}. \tag{5.10}$$

This is as far as the system can be reduced without making additional assumptions about the matrices $A$, $C$ and $F$.

A system of equation of the same form as (5.10) is called an *equilibrium system*. Equilibrium systems are commonly occurring in interior point programming. The solutions of such systems are discussed in [GL96]. In this project 5.10 is solved by the so called *range-space* method. This may not be the best solution numerically but it is both fast and works well for the tested applications. For a discussion on the numerical properties of solutions to (5.10) refer to [Vav94].

The system matrix of (5.10) is nonsingular if the matrices $A$, $C$ and $F$ has no common nullspace, that is if

$$\mathcal{N}(A)\bigcap\mathcal{N}(C)\bigcap\mathcal{N}(F) = \{0\}.$$

Unfortunately the system matrix is not positive definite. In fact the system matrix will have $q$ negative eigenvalues where $q$ is the rank of $F$. For this reason a stronger assumption is made, this is

$$\mathcal{N}(A)\bigcap\mathcal{N}(C) = \{0\}. \tag{5.11}$$

This makes the matrix $H$ positive definite which make it possible to solve (5.10) using two Cholesky factorizations. This is faster than the $LDL^T$ factorization that would be needed to solve (5.10) directly.

Note that solving (5.10) without making these assumptions will produce a perfectly valid (but not unique) step in theory. From now on it will be assumed that (5.11) hold.

From (5.10) follows that

$$H\Delta x + F^T\Delta\chi = -h$$

which implies

$$\Delta x \quad = \quad -H^{-1}F^T\Delta\chi - H^{-1}h.$$

Substituted into (5.9d) this implies

$$-FH^{-1}F^T\Delta\chi - Fh \quad = \quad -r_p$$

which simplifies to an explicit expression for $\Delta\chi$ as

$$\Delta\chi \quad = \quad (FH^{-1}F^T)^{-1}(r_p - Fh).$$

The matrix $S = FH^{-1}F^T$ is called the Schur complement of (5.10). Due to numerical considerations the inverse matrices are never explicitly formed. In the case when there are no equality constraints on $x$ system (5.10) reduce to

$$H\Delta x \quad = \quad -h$$

and $\Delta x$ is obtained as

$$\Delta x \quad = \quad -H^{-1}h.$$

System (5.2) can thus be solved by the following procedure

1. Form $D = \Lambda^{-1}V + M^{-1}W$, $D^{-1}$ and

$$r_x = r_1 + \Lambda^{-1}r_{\lambda v} - M^{-1}r_{\nu w} - \Lambda^{-1}Vr_{\xi\lambda} + M^{-1}Wr_{\xi\mu}.$$

2. Form $H = A^TA + C^TD^{-1}C$, $h = A^Tr_2 + C^TD^{-1}r_x + r_p$ and compute the Cholesky factorization of $H$.

3. Compute $H^{-1}F$ and $H^{-1}h$ by back substitution using the Cholesky factorization of $H$.

4. Form the matrix $S = FH^{-1}F^T$ and compute its Cholesky factorization.

5. Solve $S\Delta\chi = r_p - Fh$ to obtain $\Delta\chi$.

6. Compute $\Delta x$ by $\Delta x = -H^{-1}F^T\Delta\chi - H^{-1}h$.

7. Form remaining variables by

$$\begin{aligned}
\Delta\nu &= A\Delta x + r_2 \\
\Delta\xi &= D^{-1}C\Delta x + D^{-1}r_x \\
\Delta v &= \Lambda^{-1}V\Delta\xi + \Lambda^{-1}Vr_{\xi\lambda} - \Lambda^{-1}r_{\lambda v} \\
\Delta w &= -M^{-1}W\Delta\xi + M^{-1}Wr_{\xi\mu} - M^{-1}r_{\nu w} \\
\Delta\lambda &= -\Delta\xi - r_{\xi\lambda} \\
\Delta\mu &= \Delta\xi - r_{\xi\mu}.
\end{aligned}$$

### 5.2.8 Practical considerations

In order to obtain the search direction two systems of linear equations must be solved. One to obtain the affine scaling step and one for the corrector and centering step.

The most computationally expensive part of solving the system of equations are computing and factorizing the two matrices $H = A^T A + C^T D^{-1} C$ and $S = FH^{-1}F^T$. These matrices are the same for both systems and need only be computed and factorized once every iteration.

In the actual implementation equations (5.13) would be solved to obtain $\Delta z_{\text{aff}}$ as described previously. To obtain $\Delta z$ the system would not be solved using the residuals given by (5.5). Instead, since it is a linear system, it would be solved using

$$\begin{aligned} r_{\lambda v} &= \Delta\Lambda^{\text{aff}}\Delta V^{\text{aff}}\mathbf{1} - \sigma\rho\mathbf{1} \\ r_{\nu w} &= \Delta M^{\text{aff}}\Delta W^{\text{aff}}\mathbf{1} - \sigma\rho\mathbf{1} \end{aligned}$$

and all other residuals equal to 0 to obtain a correction and centering direction, $\Delta z_{\text{cc}}$. $\Delta z_{\text{cc}}$ would be added to the affine scaling direction to obtain the full search direction, $\Delta z$, as

$$\Delta z = \Delta z_{\text{aff}} + \Delta z_{\text{cc}}.$$

This of course produces the same results but requires fewer matrix vector multiplications.

## 5.3 Solving the constrained problem

As could be expected the similarity between the weighted scalarization problem and the constrained problem makes the two algorithms used to solve them very similar. In order to incorporate the constraint on the $\ell_1$-norm into the algorithm some modifications has to be made.

This section will examine these modifications and see what changes need to be made in order to deal with the added constraint. It will however not go through the whole derivation of the algorithm again since it is considered a straight forward process based on the previous section.

### 5.3.1 The modified search direction

The step equations are updated in order to incorporate the effect of the constraint on the $\ell_1$-norm. The equations corresponding to equations 5.1 are

$$\begin{aligned}
A^T\nu + C^T\xi + C^T\chi &= 0 && && \text{(5.12a)} \\
Ax - u - b &= 0 && && \text{(5.12b)} \\
Cx - v + w - d &= 0 && && \text{(5.12c)} \\
Fx - g &= 0 && && \text{(5.12d)} \\
\lambda + \xi - \eta\mathbf{1} &= 0 && && \text{(5.12e)} \\
\mu - \xi - \eta\mathbf{1} &= 0 && && \text{(5.12f)} \\
\lambda_i v_i &= \tau, && i = 1, \ldots, p && \text{(5.12g)} \\
\mu_i w_i &= \tau, && i = 1, \ldots, p && \text{(5.12h)} \\
(\eta, \lambda, \mu, v, w) &\succeq 0 && && \text{(5.12i)} \\
\eta[\alpha - \mathbf{1}^T(v + w)] &= \tau && && \text{(5.12j)} \\
\alpha - \mathbf{1}^T(v + w) &\geq 0 && && \text{(5.12k)}
\end{aligned}$$

which leads to a new definition of the function $\mathcal{F}$ as

$$\mathcal{F}(z) = \begin{bmatrix} A^T\nu + C^T\xi + F^T\chi \\ Ax - \nu - b \\ Cx - v + w - d \\ Fx - g \\ \lambda + \xi - \eta\mathbf{1} \\ \mu - \xi - \eta\mathbf{1} \\ \Lambda V\mathbf{1} \\ MW\mathbf{1} \\ \eta s \end{bmatrix}.$$

For notational convenience a parameter $s = \alpha - \mathbf{1}^T(v + w)$ has been introduced. Furthermore the notation $\Delta s$ defined by $\Delta s = -\mathbf{1}^T(\Delta v + \Delta w)$ will be used. In analogy with the previous notation $z$ denotes the collection of all variables as

$$z = (x, \nu, \xi, \chi, v, w, \lambda, \mu, \eta).$$

System (5.12) can be written as

$$\begin{array}{rcl} \mathcal{F}(z) & = & 0 \\ (\eta, s, \lambda, \mu, v, w) & \succeq & 0. \end{array}$$

Writing out the equation

$$\mathcal{J}(z)\Delta z = -\mathcal{F}(z)$$

used to obtain the search direction yields

$$\begin{bmatrix} 0 & A^T & C^T & F & 0 & 0 & 0 & 0 & 0 \\ A & -I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & -I & I & 0 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 & I & 0 & -\mathbf{1} \\ 0 & 0 & I & 0 & 0 & 0 & 0 & I & -\mathbf{1} \\ 0 & 0 & 0 & 0 & \Lambda & 0 & V & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & M & 0 & W & 0 \\ 0 & 0 & 0 & 0 & -\eta\mathbf{1}^T & -\eta\mathbf{1}^T & 0 & 0 & s \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\nu \\ \Delta\xi \\ \Delta\chi \\ \Delta v \\ \Delta w \\ \Delta\lambda \\ \Delta\mu \\ \Delta\eta \end{bmatrix} = \begin{bmatrix} 0 \\ -r_2 \\ -r_1 \\ -r_p \\ 0 \\ 0 \\ -r_{\lambda v} \\ -r_{\nu w} \\ -r_\alpha \end{bmatrix} \tag{5.13}$$

where use has been made of the fact that $r_d$, $r_{\xi\lambda}$ and $r_{\xi\mu}$ are going to be zero for all iterates. An extra residual $r_\alpha$ for the added constraint has also been introduced.

This new system can be solved by using the solution process derived in the previous section.

### 5.3.2 Changes to the centering parameter

The extra constraint added to the set of equations will require a redefinition of $\rho$. In order to incorporate 5.12j $\rho$ must be redefined as

$$\rho = \frac{\lambda^T v + \mu^T w + \eta s}{2p + 1}.$$

As before $\tau$ is given by $\tau = \sigma\rho$ where $\sigma$ is be adaptively chosen, that is

$$\sigma = \left(\frac{\rho_{\text{aff}}}{\rho}\right)^3 \tag{5.14}$$

where $\rho_{\text{aff}}$ is defined analogously to before as

$$\rho_{\text{aff}} = \frac{\lambda_{\text{aff}}^T v_{\text{aff}} + \mu_{\text{aff}}^T w_{\text{aff}} + \eta_{\text{aff}} s_{\text{aff}}}{2p+1}$$

for $z_{\text{aff}} = z + t_{\text{aff}}\Delta z_{\text{aff}}$ and

$$t_{\text{aff}} = \max\{t \in [0,1] \mid (v,w,\lambda,\mu,s,\eta) + t(\Delta v, \Delta w, \Delta\lambda, \Delta\mu, \Delta s, \Delta\eta) \succeq 0\}.$$

In analogy with the previous section $\Delta z_{\text{aff}}$ is the solution of (5.13) for $\tau = 0$.

The final step length $t$ is redefined in a similar manner as $t_{\text{aff}}$. That is

$$t = \max\{t \in [0,1] \mid (v,w,\lambda,\mu,s,\eta) + t(\Delta v, \Delta w, \Delta\lambda, \Delta\mu, \Delta s, \Delta\eta) \succeq \epsilon(v,w,\lambda,\mu,s,\eta)\} \tag{5.15}$$

for some $\epsilon > 0$.

### 5.3.3 Initial conditions and termination criteria

Some minor changes must be made to the initial parameters. The value of $v$ and $w$ must be chosen such that (5.12k) is satisfied. The initial conditions are chosen to be

$$\begin{array}{llll} x^0 = 0, & \nu^0 = 0, & \xi^0 = 0, & \chi^0 = 0, \quad \eta^0 = 1 \\ v^0 = \frac{1}{4}p^{-1}\alpha\mathbf{1}, & w^0 = \frac{1}{4}p^{-1}\alpha\mathbf{1}, & \lambda^0 = \mathbf{1}, & \nu^0 = \mathbf{1}. \end{array} \tag{5.16}$$

This choice places $\mathbf{1}^T(v+w)$ half way between 0 and $\alpha$.

The termination criteria used are

$$\frac{\|Fx - g\|_2}{1 + \|g\|_2} < \epsilon \tag{5.17a}$$

$$A^T\nu + C^T\xi + F^T\chi = 0 \tag{5.17b}$$

$$\|\xi\|_\infty \leq \gamma \tag{5.17c}$$

$$\frac{\|Ax - \nu - b\|_2}{1 + \|b\|_2} < \epsilon \tag{5.17d}$$

$$\frac{|\eta\alpha - \xi^T(Cx - d)|}{1 + \eta\alpha} < \epsilon. \tag{5.17e}$$

Thus the algorithm will stop once the iterates satisfy the optimality criteria given by (3.15) up to some relative tolerance.

### 5.3.4 Solving for the new search direction

As in the previous section the sparsity of the system matrix of (5.13) can be used in order to solve the system efficiently. In fact the same procedure as in the previous section can be used to solve large parts of the system.

First note that equation (5.13) can be rewritten as

$$\begin{bmatrix} 0 & A^T & C^T & F & 0 & 0 & 0 & 0 \\ A & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & -I & I & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 & I & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & \Lambda & 0 & V & 0 \\ 0 & 0 & 0 & 0 & 0 & M & 0 & W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta \xi \\ \Delta \chi \\ \Delta v \\ \Delta w \\ \Delta \lambda \\ \Delta \mu \end{bmatrix} = \begin{bmatrix} 0 \\ -r_2 \\ -r_1 \\ -r_p \\ 0 \\ 0 \\ -r_{\lambda v} \\ -r_{\nu w} \end{bmatrix} + \Delta\eta \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix} \tag{5.18a}$$

$$-\eta \mathbf{1}^T \Delta v - \eta \mathbf{1}^T \Delta w + s = -r_\alpha. \tag{5.18b}$$

A system similar to 5.13 is solved twice. First

$$\begin{bmatrix} 0 & A^T & C^T & F & 0 & 0 & 0 & 0 \\ A & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & -I & I & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 & I & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & \Lambda & 0 & V & 0 \\ 0 & 0 & 0 & 0 & 0 & M & 0 & W \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta \nu_1 \\ \Delta \xi_1 \\ \Delta \chi_1 \\ \Delta v_1 \\ \Delta w_1 \\ \Delta \lambda_1 \\ \Delta \mu_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -r_2 \\ -r_1 \\ -r_p \\ 0 \\ 0 \\ -r_{\lambda v} \\ -r_{\nu w} \end{bmatrix}. \tag{5.19}$$

is solved and then

$$\begin{bmatrix} 0 & A^T & C^T & F & 0 & 0 & 0 & 0 \\ A & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & -I & I & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 & 0 & 0 & I & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & \Lambda & 0 & V & 0 \\ 0 & 0 & 0 & 0 & 0 & M & 0 & W \end{bmatrix} \begin{bmatrix} \Delta x_2 \\ \Delta \nu_2 \\ \Delta \xi_2 \\ \Delta \chi_2 \\ \Delta v_2 \\ \Delta w_2 \\ \Delta \lambda_2 \\ \Delta \mu_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix}. \tag{5.20}$$

Both these systems of the same form as the system shown in the previous section and is therefore easily solved. By linearity the solution to (5.13) can be written as $\Delta z = \Delta z_1 + \Delta\eta\Delta z_2$ and in particular

$$\Delta v = \Delta v_1 + \Delta\eta\Delta v_2$$
$$\Delta w = \Delta w_1 + \Delta\eta\Delta w_2.$$

Inserting this into equation 5.18b result in

$$-\eta \mathbf{1}^T(\Delta v_1 + \Delta\eta\Delta v_2) - \eta \mathbf{1}^T(\Delta w_1 + \Delta\eta\Delta w_2) + \Delta\eta s = -r_\alpha$$

which implies

$$\Delta\eta = \frac{r_\alpha - \eta \mathbf{1}^T(\Delta v_1 + \Delta w_1)}{\eta \mathbf{1}^T(\Delta v_2 + \Delta w_2) - s}.$$

Using the notation $\Delta s = -\mathbf{1}^T(\Delta v + \Delta w)$ the above expression simplifies to

$$\Delta\eta = -\frac{r_\alpha + \eta \Delta s_1}{\eta \Delta s_2 + s} \tag{5.21}$$

The full solution to equation (5.13) can thus be obtained as

$$\begin{bmatrix} \Delta x \\ \Delta\nu \\ \Delta\xi \\ \Delta\chi \\ \Delta v \\ \Delta w \\ \Delta\lambda \\ \Delta\mu \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \Delta\nu_1 \\ \Delta\xi_1 \\ \Delta\chi_1 \\ \Delta v_1 \\ \Delta w_1 \\ \Delta\lambda_1 \\ \Delta\mu_1 \end{bmatrix} + \Delta\eta \begin{bmatrix} \Delta x_2 \\ \Delta\nu_2 \\ \Delta\xi_2 \\ \Delta\chi_2 \\ \Delta v_2 \\ \Delta w_2 \\ \Delta\lambda_2 \\ \Delta\mu_2 \end{bmatrix}.$$

### 5.3.5 Detecting infeasibility

One drawback of this scheme is that it does not provide an explicit way for detecting infeasibility. This is however only an issue when $d \notin \text{range}(C)$ since otherwise 0 is a tight lower bound on $\|Cx - d\|_1$ and detecting infeasibility corresponds to determining if $\alpha$ is negative.

Since all applications considered in this thesis satisfy $d \in \text{range}(C)$ this will not be investigated further. If however this would be needed for some application not covered here the minimum of $\|Cx - d\|_1$ over all $x$ such that $Fx = g$ could be efficiently computed by a linear program.

### 5.3.6 Practical considerations

Note that system (5.20) is the same when calculating the affine scaling step and the centering and correction step. For this reason it is only needed to solve this system once per iteration. Thus three system of equations of the form (5.2) must be solved at each iteration. As noted in the previous section since the coefficient matrix is the same for all three systems, this does not increase the number of operations significantly.

## 5.4 Algorithm summary

### 5.4.1 The weighted scalarization problem

1. Initialize $z$ according to (5.7).
2. If $z$ satisfies the termination criteria given by (5.8) exit and return $z$.
3. Form matrices $H = A^T A + C^T D^{-1} C$ and $S = F H^{-1} F^T$ and compute their Cholesky factorizations. These factorizations are used to solve the systems of linear equations in step 4 and 6.
4. Let
$$\begin{aligned} r_2 &= Ax - b - \nu \\ r_1 &= Cx - v + w - d \\ r_p &= Fx - g \\ r_{\lambda v} &= \Lambda V \mathbf{1} \\ r_{\mu w} &= MW\mathbf{1} \end{aligned}$$
where the remaining residuals are equal to zero and solve (5.2) in order to obtain affine scaling direction $\Delta z_{\text{aff}}$.
5. Compute the constant $\rho$ and the centering parameter $\sigma$ according to (5.4).
6. Let
$$\begin{aligned} r_{\lambda v} &= \Delta\Lambda^{\text{aff}} \Delta V^{\text{aff}} \mathbf{1} - \sigma\rho\mathbf{1} \\ r_{\nu w} &= \Delta M^{\text{aff}} \Delta W^{\text{aff}} \mathbf{1} - \sigma\rho\mathbf{1} \end{aligned}$$
where the remaining residuals are equal to zero and solve (5.2) in order to obtain the correction and centering direction $\Delta z_{\text{cc}}$.
7. Form the search direction as $\Delta z = \Delta z_{\text{aff}} + \Delta z_{\text{cc}}$ and compute the step length $t$ according to (5.6).
8. Update the iterate as $z := z + t\Delta z$ and go back to step 2.

### 5.4.2 The constrained problem

1. Initialize $z$ according to (5.16).
2. If $z$ satisfies the termination criteria given by (5.17) exit and return $z$.
3. Form matrices $H = A^T A + C^T D^{-1} C$ and $S = F H^{-1} F^T$ and compute their Cholesky factorizations. These factorizations are used to solve the systems of linear equations in step 4, 5 and 7.
4. Solve 5.20 in order to obtain $\Delta z_2$. The solution is obtained by solving (5.2) for

$$\begin{aligned} r_{\xi\lambda} &= -\mathbf{1} \\ r_{\xi\mu} &= -\mathbf{1} \end{aligned}$$

   where the remaining residuals are equal to zero.
5. Let

$$\begin{aligned} r_2 &= Ax - b - \nu \\ r_1 &= Cx - v + w - d \\ r_p &= Fx - g \\ r_{\lambda v} &= \Lambda V \mathbf{1} \\ r_{\mu w} &= MW\mathbf{1} \end{aligned}$$

   where the remaining residuals are equal to zero and solve (5.19) in order to obtain $\Delta z_1$. Compute $\Delta\eta$ by (5.21) and compute the affine scaling direction $\Delta z_{\text{aff}}$ as $\Delta z_{\text{aff}} = \Delta z_1 + \Delta\eta \Delta z_2$.
6. Compute the constant $\rho$ and the centering parameter $\sigma$ according to (5.14).
7. Let

$$\begin{aligned} r_{\lambda v} &= \Delta\Lambda^{\text{aff}} \Delta V^{\text{aff}} \mathbf{1} - \sigma\rho\mathbf{1} \\ r_{\nu w} &= \Delta M^{\text{aff}} \Delta W^{\text{aff}} \mathbf{1} - \sigma\rho\mathbf{1} \end{aligned}$$

   where the remaining residuals are equal to zero and solve (5.19) in order to obtain $\Delta z_1$. Compute $\Delta\eta$ by (5.21) and compute the correction an centering direction $\Delta z_{\text{cc}}$ as $\Delta z_{\text{cc}} = \Delta z_1 + \Delta\eta \Delta z_2$.
8. Form the search direction as $\Delta z = \Delta z_{\text{aff}} + \Delta z_{\text{cc}}$ and compute the step length $t$ according to (5.15).
9. Update the iterate as $z := z + t\Delta z$ and go back to step 2.

# Chapter 6

# Applications

In this chapter several applications of the bi-criterion problem will be presented. These applications rely on the solution of either single objective reformulations.

In the applications one of the two single objective reformulations will be more natural. Some of the applications will be very similar mathematically.

It is not within the scope of this report to give a detailed description of each application. The sections of this chapter intend to give a sufficient and intuitive explanation of possible applications of the $\ell_1/\ell_2$ bi-criterion problem. The application examples include a brief description of the application along with a numerical example. The numerical examples are included to illustrate the characteristics of the application along with a discussion of the effectiveness of the algorithms presented in chapter 5.

## 6.1 Numerical results

All numerical results presented in this chapter was obtained using `MATLAB` functions `l1l2optw` and `l1l2optc` (described in appendix A). These functions are `MATLAB` implementations of the algorithms presented in chapter 5.

The simulations were performed on a system running Red Hat Linux 7.2 and `MATLAB` 6.0 with a 1.3GHz AMD Athlon processor and 1 Gb main memory. The timing results presented are intended to give a rough estimate of the computational complexity of the problems. Some effort has been made in order to ensure that the algorithms were running in main memory at close to 100% cpu usage.

## 6.2 The LASSO

### 6.2.1 Linear regression

In a linear regression problem one wish to form a linear estimate of a quantity from the measurements of $n$ different quantities. In order to do this an experiment or measurement are repeated $m$ times. One thus obtain data of the form $(x_i, y_i)$ where $x_i \in \mathbf{R}^n$ and $y_i \in \mathbf{R}$. Here $x_i$ are called the *regressors* and $y_i$ the *response* of the $i$th measurement. The object is to find a $\beta \in \mathbf{R}^n$ such that

$$y_i = x_i^T \beta, \qquad i = 1 \dots m$$

or equivalently

$$y = X\beta \tag{6.1}$$

where $y \in \mathbf{R}^m$ and $X = [x_1, \dots, x_n]^T$. Thus $\beta$ will form a linear functional that serves as the predictor.

Due to measurement noise and nonlinearity this system will in general not have a solution for $m > n$. In these cases it is better to obtain the least squares estimate of $\beta$.

The least squares estimate has the lowest variance in the estimate of all linear unbiased estimates under a white Gaussian noise assumption [HTF01]. Nevertheless when the matrix $X$ is poorly conditioned the estimate $\hat{\beta}$ of $\beta$ will have large variance. This has motivated the development of unbiased estimators that trade off some of the bias for a better variance. These include *subset selection* and *shrinkage methods*. In subset selection one chose $k$ components of $\beta$ for $k < n$ to be used for the predictor and set the rest to zero. In the shrinkage methods one instead introduce some form of penalty on unwanted estimates $\hat{\beta}$. The estimates are then obtained as the optimal point of

$$\text{minimize} \quad \|y - X\beta\|_2^2 + \lambda f(\beta)$$

where the penalty $f(\beta)$ is generally a convex function and $\gamma$ is a positive constant. With the choice of $f(\beta) = \|\beta\|_2^2$ this leads to the convex problem

$$\text{minimize} \quad \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$$

which in statistics is called *ridge regression*.

### 6.2.2 LASSO estimates

Proposed by Tibshirani [Tib96] the LASSO, *Least Absolute Shrinkage and Selection Operator*, is a form of shrinkage method that put an $\ell_1$ norm constraint on the estimate. The LASSO estimate is obtained by solving

$$\begin{array}{ll} \text{minimize} & \|y - X\beta\|_2^2 \\ \text{subject to} & \|\beta\|_1 \leq t \end{array} \tag{6.2}$$

over $\beta \in \mathbf{R}^n$ for some $t \geq 0$. By the nature of the $\ell_1$ norm the solution of (6.2) tends to be sparse. Due to this the LASSO can also be viewed as a subset selection method.

Sparsity of $\hat{\beta}$ is a desirable property since it will usually give useful information about the measurements. Components of $x_i$ that correspond to a zero in $\beta$ does not affect the output $y_i$. The

LASSO estimate will thus give information as to which quantities actually have a significant effect on the output and enable simplification of the model by incorporating only these quantities.

A proper statistical analysis of the LASSO and comparisons to other shrinkage and subset selection methods are given in [HTF01]. For the purpose of this project it is sufficient to note that the LASSO problem give rise to a problem that falls into the class of bi-criterion $\ell_1/\ell_2$-norm problems discussed in this project.

### 6.2.3 A numerical example

#### Problem setup

This example is taken from the original paper on the LASSO by Tibshirani [Tib96]. In this example the correlation between the level of prostate specific antigen and a set of clinical measurements are examined. In the example a linear estimate of the log of prostate specific antigen is made from some clinical measurements after these are normalized to have zero mean and unit variance.

The regression variables (clinical measurements) were, log cancer volume (lcavol), log prostate weight (lweight), age, log of benign prostate hyperplasia amount (lbph), seminal vesicle invasion (svi), log capsular penetration (lcp), gleason score (gleason) and percentage gleason score 4 or 5 (pgg45).

The data represents the measurements of the above regression variables along with the response for 97 different patients. For a more complete analasys of the data refer to [Tib96].

#### Solution methodology

The regressors $x_i$ and responses $y_i$ are collected in a dense matrix $X$ of size 97 by 8 and a vector $y$ of length 97 respectively. The regression variables were obtained using `l1l2optc` for 100 values of $\alpha$ equally distributed from 0 to 1.1 times $\alpha_{\max}$, where $\alpha_{\max}$ is the smallest value of $\alpha$ that gives the least squares solution. These are shown in figure 6.1

For an $\alpha$ of 44% of $\alpha_{\max}$ `l1l2optc` converge to a relative tolerance of $10^{-8}$ in 7 iterations. The solution is obtained in 0.014 seconds.

#### Results

Figure (6.1) shows how the parameters depend on $\alpha$. For values of $\alpha$ greater than 1.84 the least squares estimate of the parameters is obtained. From this plot it is easily seen which parameters will have a larger effect on the prediction. If the experiment were to be repeated with only 3 regression variables lcavol, svi and lweight are the variables that should be chosen.

The $\alpha$ used by Tibshirani is 0.44 percent of the value of $\alpha$ that correspond to the least squares solution, *i.e.* $\alpha = 0.81$. In [Tib96] this $\alpha$ is obtained through a process called cross validation. For this value of $\alpha$ the solution given by table 6.1 is obtained. The values in this table are the same as those obtained by Tibshirani and are equal up to at least 4 decimals to the values obtained and presented by Osborne *et. al.* in [OPT98].

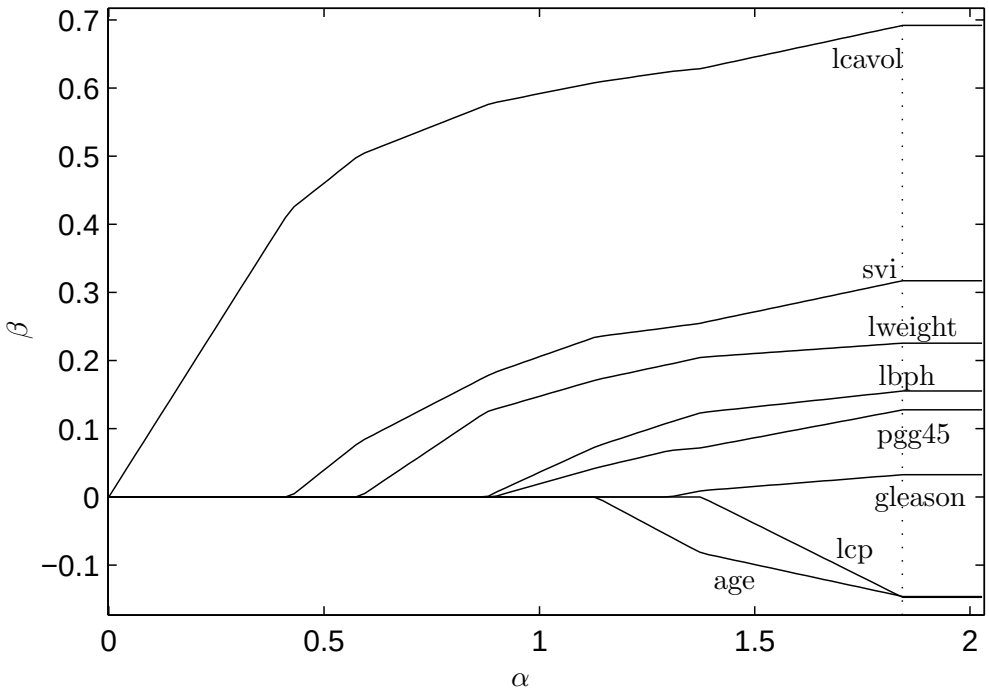


**Figure 6.1:** LASSO coefficients as a function of problem parameter $\alpha$. The dotted line represent the $\alpha$ for which we obtain the least square solution.

| Predictor | Coefficient |
|---|---|
| lcavol | 0.5588 |
| lweight | 0.0970 |
| age | 0.0000 |
| lbph | 0.0000 |
| svi | 0.1556 |
| lcp | 0.0000 |
| gleason | 0.0000 |
| pgg45 | 0.0000 |

**Table 6.1:** Coefficients obtained by solving the LASSO problem of the prostate cancer example at $\alpha = 0.81$.

### 6.2.4 Related material

The LASSO and the solution thereof has been studied by Osborne *et. al.* in [OPT98]. This paper contains formulation of the dual problem of the LASSO and the development of an algorithm for the solution of the problem based on this dual.

A primal dual algorithm for the solution of a generalized version of the LASSO problem is presented in [TVW01].

## 6.3 Basis pursuit

### 6.3.1 Overcomplete dictionaries

This section will deal with the problem of representing a signal as a sum of weighted basis signals, *i.e.* finding a representation of a signal in some given dictionary. This is for example what is done when one the discrete Fourier transform of a signal is computed. In the case of Fourier transforms the signal dictionary is orthogonal. This section deal with signal dictionaries where this is not the case. In fact the dictionaries considered here are overcomplete.

Overcomplete in this context means that there are more signals in the dictionary than the dimension of the signal to be represented in the dictionary. This means there will be many ways of representing the given signal. Given a dictionary $\mathcal{D}$, which is a collection of signals $\phi_i$, and a signal $s$ the object is to obtain parameters $\alpha_i$ such that

$$s = \sum_i \alpha_i \phi_i + r$$

where $r$ is some residual.

In many cases it would be desirable to represent $s$ as a sum of as few signals $\phi_i$ as possible, either because a sparse solution will be easier to store or because a sparse solution will provide important information about the nature of $s$. Ideally the residual $r$ would be zero but this section will consider the case were some of the residual can be traded off for increased sparsity.

### 6.3.2 Basis pursuit

Basis pursuit (BP) [CDS99] can be viewed as a heuristic for obtaining a sparse representation of a signal in some overcomplete dictionary. The BP principle is to choose the dictionary that minimizes the $\ell_1$ norm of the signal coefficients. That is to choose $\alpha$ such that it is the solution of

$$\begin{array}{ll} \text{minimize} & \|\alpha\|_1 \\ \text{subject to} & \Phi\alpha = s \end{array} \tag{6.3}$$

over $\alpha \in \mathbf{R}^n$ where $\Phi = [\phi_1 \dots \phi_n]$.

In the case of noisy data it may be more suitable to replace the equality constraint by a quadratic penalty on the residual $r = s - \Phi\alpha$.

Basis pursuit denoising (BPDN) refers to the problem

$$\text{minimize} \quad \tfrac{1}{2} \|s - \Phi\alpha\|_2^2 + \gamma \|\alpha\|_1 . \tag{6.4}$$

BPDN can of course be applied even in the absence of noise and be viewed as a tradeoff between perfectly fit of the signal and sparsity of the solution. BPDN will be used as a heuristic to obtain a sparsity pattern. For this reason the solution, $\alpha^\star$, of

$$\begin{array}{lll} \text{minimize} & \|s - \Phi\alpha\|_2^2 & \\ \text{subject to} & \alpha_i = 0 & \text{if } a'_i = 0 \end{array} \tag{6.5}$$

where $\alpha'$ is the solution of (6.4) will be the final solution of the BPDN problem.

### 6.3.3 A numerical example

#### Problem setup

The goal of this example is to obtain a sparse representation of the signal

$$x(t) = a(t)\sin\theta(t) \tag{6.6}$$

where

$$a(t) = 1 + 0.5\sin(11t)$$

and

$$\theta(t) = 30\sin(5t).$$

This signal is chosen for no reason other than that it exhibits noticeable changes in its spectrum over time. The signal is shown in figure 6.2. $a(t)$ can be viewed as the signal amplitude and $\theta(t)$ as the signal phase. Furthermore the concept of *instantaneous frequency* $\omega(t)$ given by

$$\omega(t) = \left|\frac{d\theta}{dt}\right| = 150|\cos(5t)|. \tag{6.7}$$

is introduced.

The object will be to find a sparse representation of this signal in a dictionary indexed by two indexes, $s$ and an integer $k$. The dictionary signals are given by

$$\phi_{s,k} = \begin{cases} e^{\frac{(x-s)^2}{\sigma^2}} & \text{for } k = 0 \\ e^{\frac{(x-s)^2}{\sigma^2}}\sin(\frac{k+1}{2}\omega_0 t) & \text{for } k > 0,\ k \text{ odd} \\ e^{\frac{(x-s)^2}{\sigma^2}}\cos(\frac{k}{2}\omega_0 t) & \text{for } k > 0,\ k \text{ even} \end{cases}, \tag{6.8}$$

*i.e.* ordinary sine and cosine functions windowed by a Gaussian function. These basis functions are called Gabor functions and are commonly used for time-frequency analysis of signals.

The signal $x(t)$ and the dictionary signals are sampled on the interval $t \in [0, 1]$ with a sample time $T$ of 0.002 which makes them 501 samples long. A dictionary is constructed by choosing $s$ to match these sample points and $k$ to take on integer values between 0 and 60. The dictionary will therefore contain $501 \times 61 = 30561$ signals.

The width of the Gaussian function, $\sigma$, is chosen to be 0.05. The basis frequency $\omega_0$ is chosen to be 5. The reason for this is that the maximum frequency of the sinusoidal parts of the dictionary signal match the highest possible value of the instantaneous frequency for this choice of $\omega_0$. To simplify the computations the basis functions are truncated such that components of the basis signal, where the Gaussian term is less than 0.001, is set to zero. For the above choice of $\sigma$ this implies that there will be at most 131 non-zeros components in each signal within the dictionary. A few signals from the dictionary are shown in figure 6.3.

#### Solution methodology

The construction of this dictionary leads to a matrix, $\Phi$, of dimension 501 by 30561. Due to the truncation of the dictionary signals this matrix has 3739821 nonzero entries. This matrix is

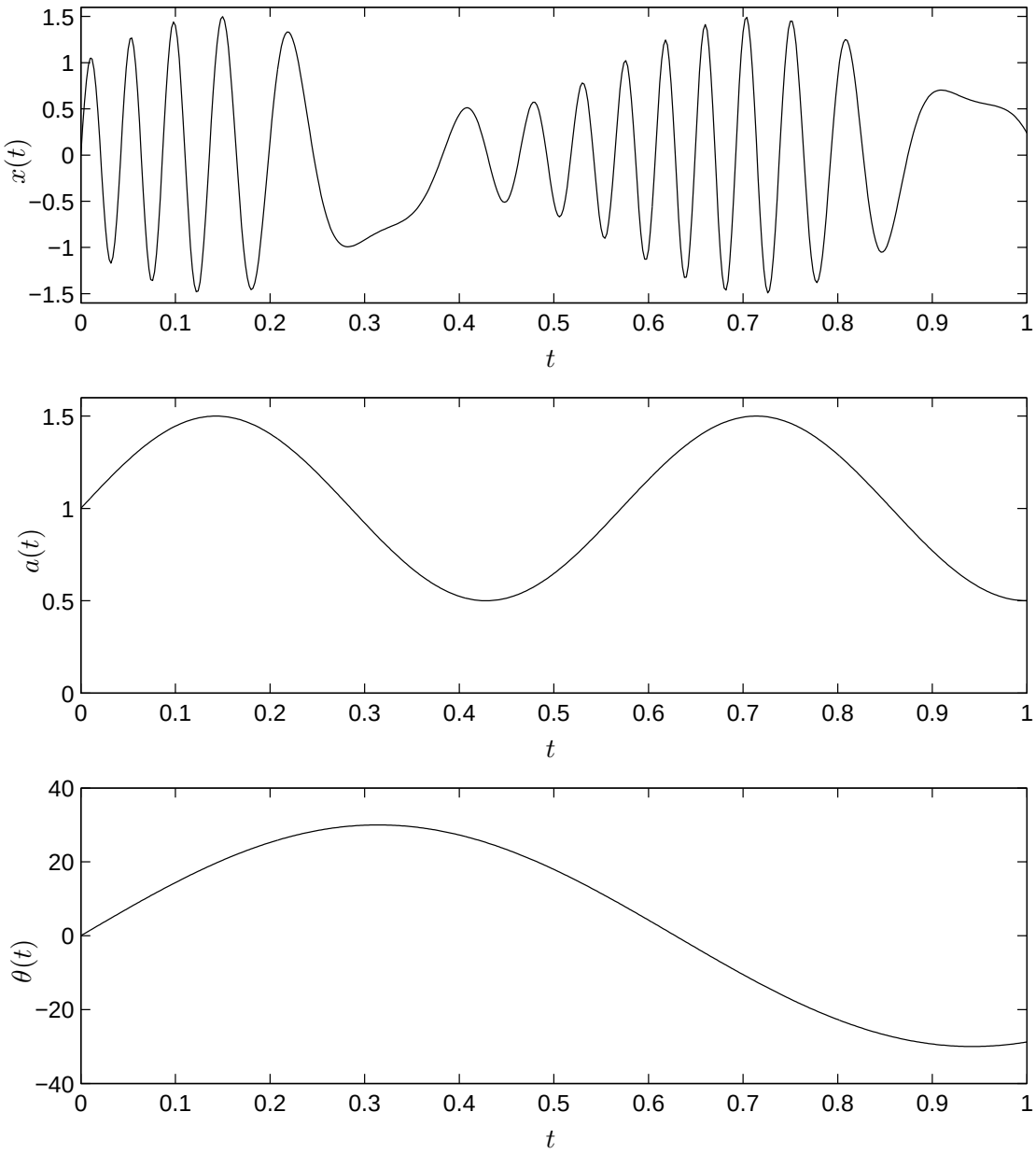


**Figure 6.2:** Signal $x(t)$ with rapidly varying spectral properties. The amplitude $a(t)$ and phase $\theta(t)$ are shown as well as the signal $x(t)$.

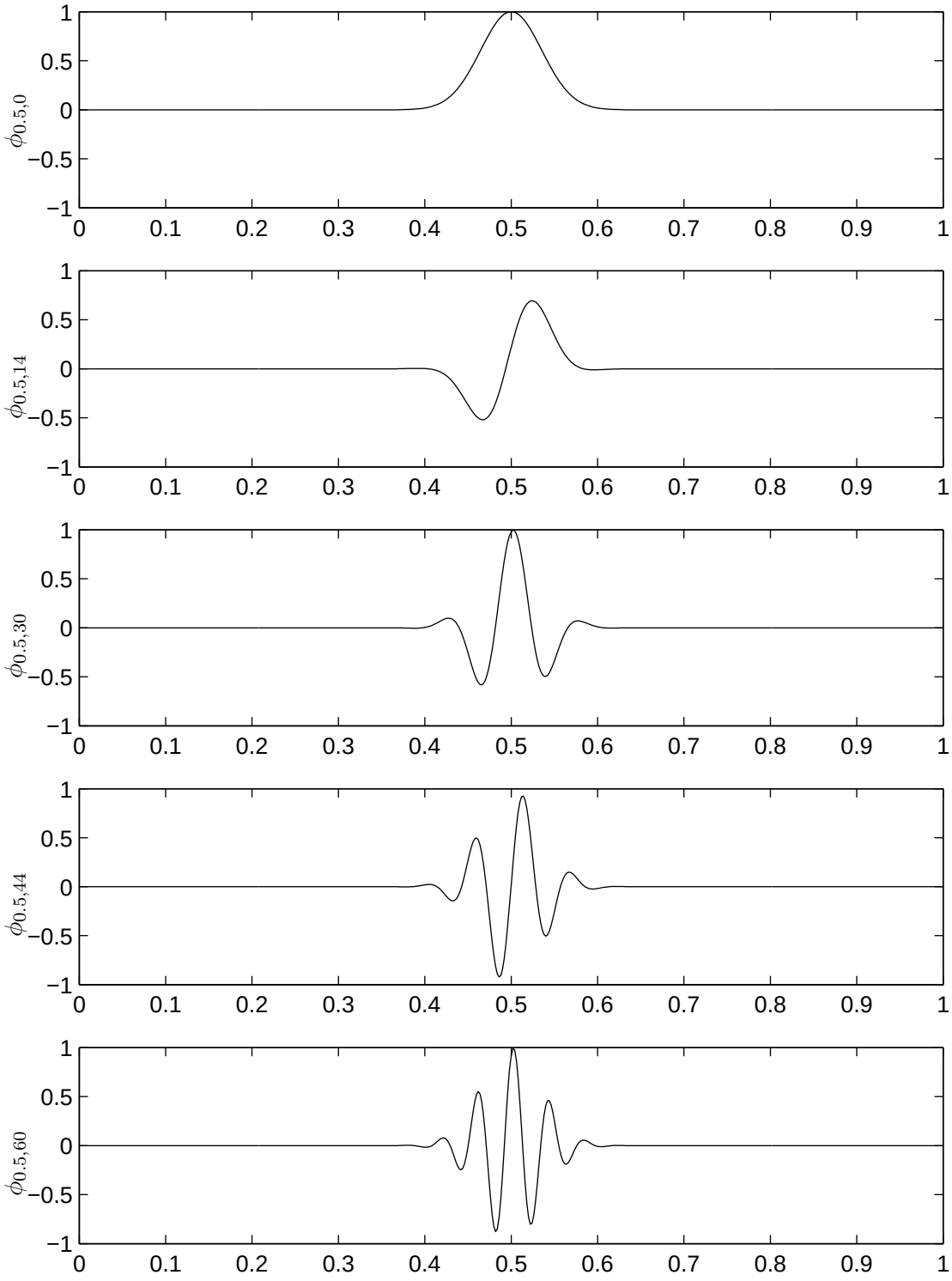


**Figure 6.3:** A sample of dictionary signals for $s = 0.5$.

constructed and the BPDN solution is obtained for $\gamma = 1$ using a modified version of the `MATLAB` function `l1l2optw` called `l1l2optwbp`.

The `MATLAB` function `l1l2optwbp` is the same as `l1l2optw` except in the way it solves the equation used to obtain the step direction. This equation is on the form

$$(\Phi^T\Phi + D^{-1})\Delta x = -h$$

where $D$ is a diagonal matrix and $h$ is a vector in $\mathbf{R}^{30561}$. This is an equation of 30561 unknowns. Due to the special structure of this system it can be solved more efficiently by first obtaining $y$ as the solution of

$$(I + \Phi D\Phi^T)y = -ADh.$$

which is a system of only 501 unknowns. $\Delta x$ can then be computed as

$$\Delta x = -D(h + \Phi^T y).$$

This trick will significally decrease the number of operations needed in this particular problem. It should be easily verified that this $\Delta x$ solves the original system.

In a real basis pursuit application the object would be to heavily utilize the structure of the chosen basis. The only structure `l1l2optwbp` takes advantage of is the sparsity of the dictionary matrix $\Phi$, but this is sufficient to solve the problem in an acceptable time for this example.

The algorithm converge to a tolerance of $10^{-8}$ in 20 iterations and takes a total of 404 seconds to complete, *i.e.* just under 7 minutes.

### Results

The solution $\alpha^\star$ obtained has 42 nonzero entries out of 30561. For this $\alpha^\star$ the signal representation has a relative accuracy of

$$\frac{\|x - \Phi\alpha^\star\|_2^2}{\|x\|_2^2} = 2.6 \times 10^{-4}.$$

The original signal along with the sparse representation is shown in figure 6.4.

The nonzero entries of $\alpha^\star$ are shown in figure 6.5. The nonzero components are chosen by the algorithm such that their frequency closely match the instantaneous frequency for the corresponding value of $t$.

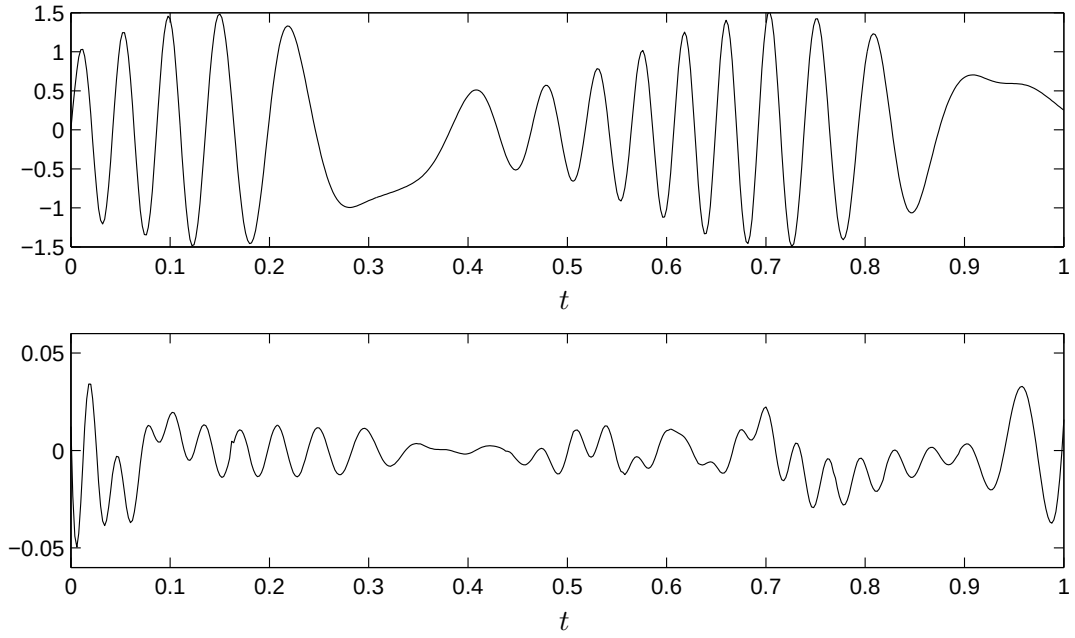


**Figure 6.4:** Signal representation in the given basis. The figure show the sparse reconstruction of the signal in the given basis. The bottom plot show the reconstruction error.

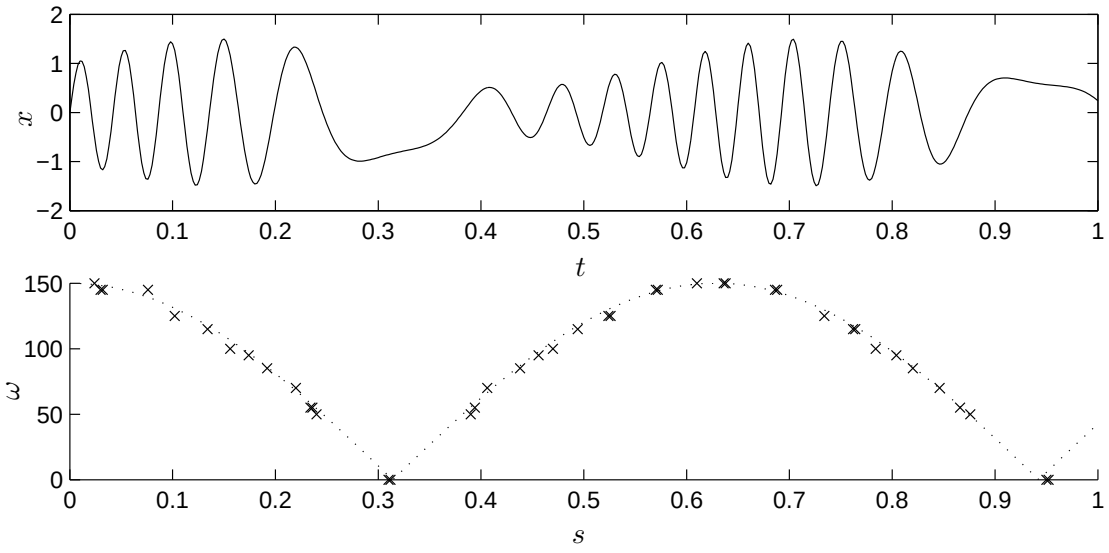


**Figure 6.5:** Frequency of basis components as a function of $s$. A cross in the second plot indicates a nonzero entry in $a^\star$ that corresponds to a given time shift $s$ and frequency $\omega$. The instantaneous frequency $\omega(t)$ is shown as a reference.

## 6.4 Total variation denoising

### 6.4.1 Regularization

This example a way to obtain an estimate $\hat{x}$, of a signal $x \in \mathbf{R}^n$, from an observed signal $y = Ax+n$ where $n \in \mathbf{R}^m$ is an additive noise. Ignoring the noise and solving the system $y = Ax$ is generally a bad idea, especially if $A$ is close to singular. Such a solution will generally tend to amplify the noise.

A way of producing a more stable estimate $\hat{x}$ is to use a regularization function $\phi(x)$ and solve

$$\text{minimize} \quad \tfrac{1}{2}\|Ax - y\|_2^2 + \gamma\phi(x). \tag{6.9}$$

A common choice of $\phi(x)$ is $\phi(x) = \|x\|_2^2$ which penalize the solution for having large variance. This is called Tikhonov regularization.

If it is known a-priori that the signal $x$ is smooth, *i.e.* has less energy in the high frequency components, a quadratic smoothing function defined as

$$\phi_{quad}(x) = \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2. \tag{6.10}$$

can be used. Using (6.10) as the regularization function in (6.9) is equivalent to applying Tikhonov regularization to the difference of adjacent samples of the signal. This will heavily penalize signals with rapid variation and produce a smooth result.

### 6.4.2 Total variation

Total variation (TV), introduced by Rudin *et. al.* [ROF92], is a noise removal scheme for signals that exhibit discontinuities. Such signals will be poorly recovered using the quadratic smoothing regulation $\phi_{quad}(x)$ as this function will heavily penalize non smooth signals.

Total variation works as the quadratic smoothing but uses an $\ell_1$-norm instead of the squared $\ell_2$-norm. More precisely TV denoising solves (6.9) with the regularization function given by

$$\phi_{TV}(x) = \sum_{i=1}^{n-1}|x_{i+1} - x_i|. \tag{6.11}$$

Solving (6.9) with (6.11) will tend to produce piecewise constant solutions.

Figure (6.6) shows the difference between quadratic smoothing and total variation. As can be seen in the figure when applied to a signal with sharp transition the quadratic smoothing tend to smooth out these transitions. The total variation reconstruction will leave these transitions alone.

This of course goes two ways. If the original signal is smooth then total variation will still return a signal that tends to be blocky while the quadratic smoothing will produce a more satisfactory result.

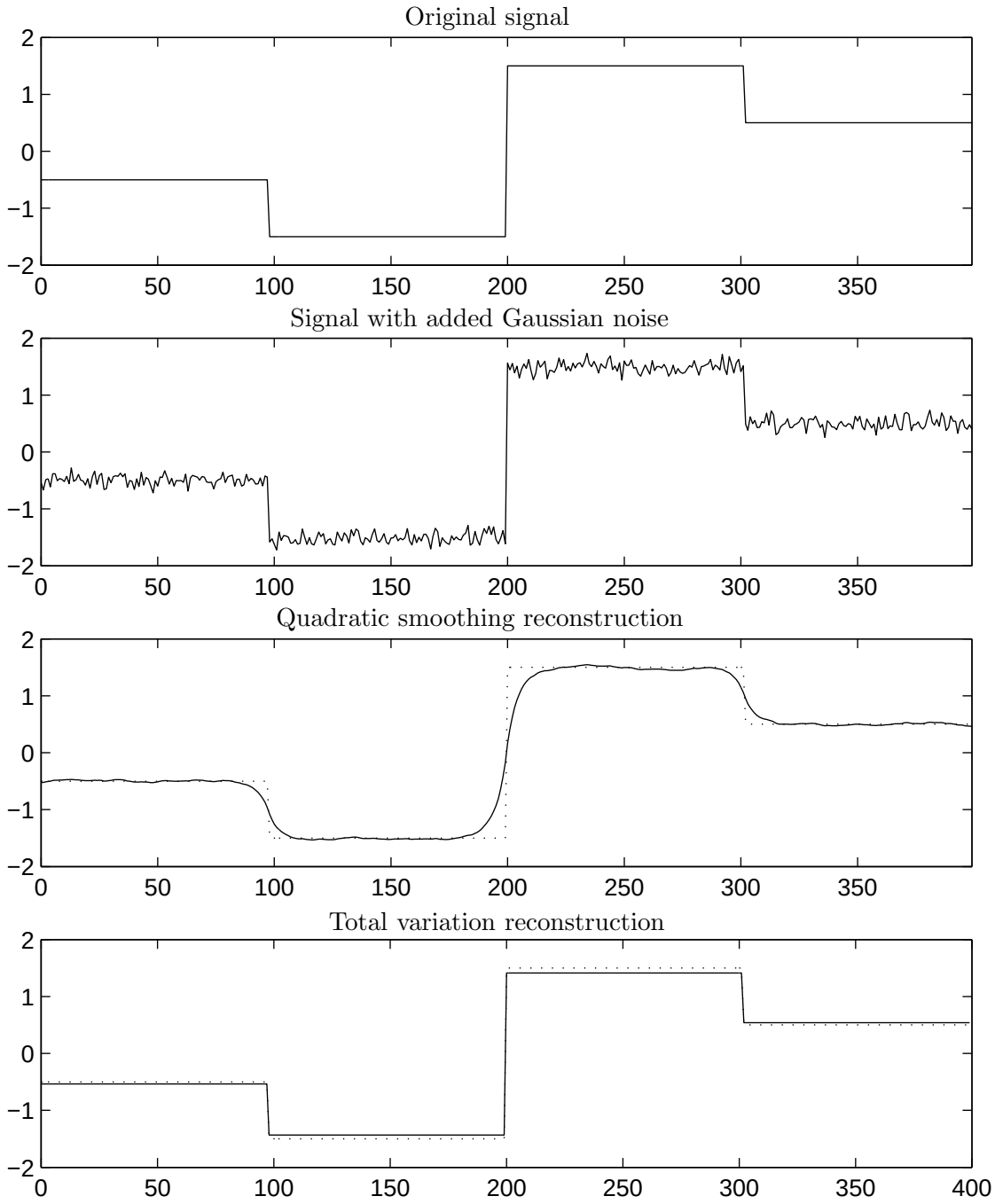


**Figure 6.6:** Comparison of quadratic smoothing reconstruction and total variation reconstruction applied to a piecewise constant signal. In the lower two plots the original signal is shown as a dotted line.

### 6.4.3 Total variation image reconstruction

An area where TV has attracted much attention is in image denoising. The reason for this is that an image, while smooth in large regions, will generally have a few sharp transitions (edges).

We will let an image $x$ consist of pixels $x_{i,j}$ where $i$ and $j$ denotes the row and column of the pixel. The most common regularization function in total variation denoising of images is (see [CGM99])

$$\phi_{TV}(x) = \sum_{i,j} \sqrt{(x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2} \tag{6.12}$$

which is a rotationally invariant estimation of the total variation of the image.

While regularization using this function does not fall within the scope of this project a related function introduced and analyzed by Li and Santosa [LS96] does. This function is defined by

$$\phi_{TV}(x) = \sum_{i,j} |x_{i,j+1} - x_{i,j}| + |x_{i+1,j} - x_{i,j}|. \tag{6.13}$$

In their paper [LS96] Li and Santosa develop an algorithm for solving (6.9) using the regularization function given by (6.13). They also present results for a variety of different matrices $A$ corresponding to different levels of image blurring.

### 6.4.4 A numerical example

#### Problem setup

Total variation denoising is used to remove noise from the $512 \times 512$ Lena image shown in figure 6.7.

The gray-scale pixels of this image are normalized to $[0, 1]$ where 0 correspond to black and 1 to white. A white Gaussian noise of variance $\sigma^2$ is added to the image. For this example $\sigma = 0.05$.

The noisy image is denoted $y$ and the denoised image $\hat{x}$ is obtained as the solution of

$$\text{minimize} \quad \textstyle\sum_{i,j} (x_{i,j} - y_{i,j})^2 + \gamma \phi_{TV}(x)$$

where $\phi_{TV}$ is given by (6.13). The value of $\gamma$ chosen for the example is 0.03. The reason for this choice of $\gamma$ is no other than that it provides visually pleasing results.

#### Solution methodology

The noisy image is converted into a vector $y$ of length 262144 where each component correspond to a pixel in the image.

A matrix $D$ is created to form the difference between pixels both horizontally as well as vertically. This matrix $D$ is a highly sparse matrix of size 523264 by 262144. Each row of $D$ has only 2 nonzero components, a $+1$ and a $-1$. Using $D$ $\phi_{TV}(x)$ can be written as

$$\phi_{TV}(x) = \|Dx\|_1 .$$

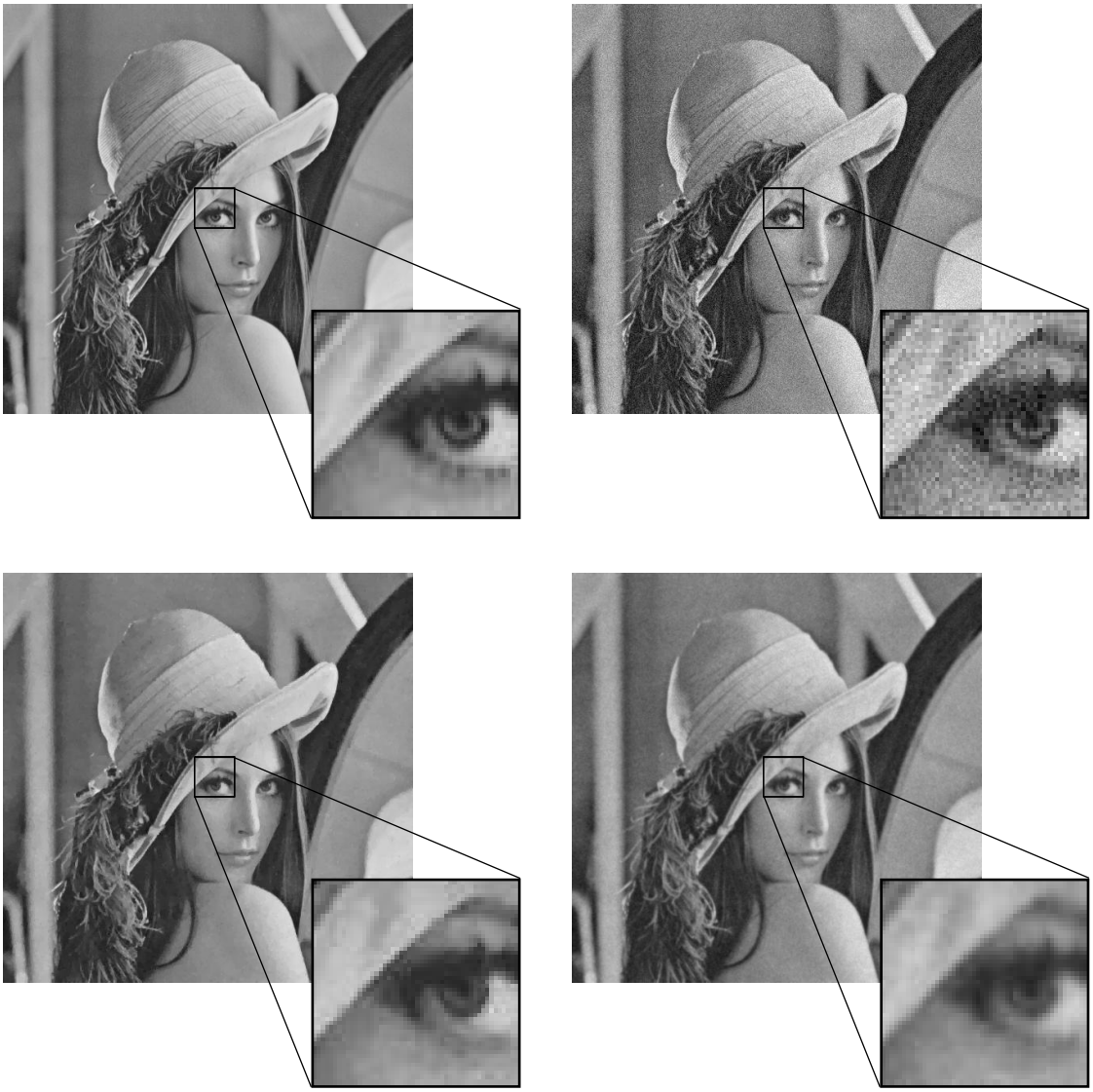

**Figure 6.7:** Total variation denoising of $512 \times 512$ Lena image. The top left image shows the original image while the top right show the noisy image. The lower left image is the result of total variation denoising. For comparison the lower right image show the result as obtained by quadratic smoothing.

The problem can now be written as

$$\text{minimize} \quad \|x - y\|_2^2 + \gamma \|Dx\|_1 .$$

This problem is solved using `l1l2optw` for the specified value of $\gamma$. The problem is a optimization problem of 262144 variables, but due to the sparsity of $D$ it is readily solved. The function `l1l2optw` converge to a tolerance of $10^{-8}$ in 15 iterations. This is accomplished in a total of 1145 seconds, *i.e.* roughly 19 minutes.

**Results**

The results of the total variation denoising are shown in figure (6.7). As a comparison the results obtained by quadratic smoothing are also shown. This refers to the solution of

$$\text{minimize} \quad \|x - y\|_2^2 + \gamma_{\text{ls}} \|Dx\|_2^2$$

for $\gamma_{\text{ls}} = 1.0$. This value was chosen since it gave, at least visually, an equal noise reduction to the total variation denoising.

Using the same method an image can be denoised and deblured for an arbitrary matrix $A$. This however come at an increased cost of computation when the matrix $A$ is not as sparse as $D$.

## 6.5 Robust linear estimation

### 6.5.1 Linear estimation

This section deals with approximately solving an overcomplete system $Ax = b$ by minimizing some residual vector

$$r = Ax - b$$

over $x \in \mathbf{R}^n$ where $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$ and $\mathbf{Rank}\, A = n$.

This problem is the same as the one initially considered in the LASSO section. By far the most common solution to the problem is to minimize the Euclidean norm of $r$. That is to obtain the *least-squares* estimate of $x$ by solving

$$\text{minimize} \quad \|Ax - b\|_2^2 . \tag{6.14}$$

The least squares, while being easy to compute, has some shortcomings. Consider the system where $b$ is given by

$$b = Ax + n$$

where $x \in \mathbf{R}^n$ are problem parameters one which to estimate and $n \in \mathbf{R}^m$ some noise. As pointed out in the LASSO section if $n$ is white Gaussian noise it can be shown that the least-squares estimator is the unbiased estimator with the smallest estimation variance.

A problem with the least-squares estimator is the quadratic penalty of components in $r$. If $n$ is so called *shot-noise* with a few very large components these are going have a large effect on the estimate. If $b$ corresponds to some measured quantity the measurement $b_i$ that correspond to a large value of $n_i$ is called an *outlier*. Figure (6.8) shows a least squares line fit to a set of points with an outlier $b_i$. As shown in the figure such an outlier can cause a bad fit due to the fact that the square function heavily penalizes the residual $r_i$ corresponding to $b_i$.

### 6.5.2 Robust linear regression

A popular solution to this problem is to use the Huber M-estimator cost function [Hub81], $\rho(t)$, defined by

$$\rho(t) = \begin{cases} \frac{1}{2}t^2, & \text{if } |t| \leq \gamma \\ \gamma|t| - \frac{1}{2}\gamma^2 & \text{if } |t| > \gamma. \end{cases} \tag{6.15}$$

This function is an ordinary quadratic function for a parameter $t$ smaller than some $\gamma$ and linear for values of $t$ greater than this threshold. The function can thus be viewed as a quadratic penalty function for small values of the residual while only assigning linear penalty to outliers. This is shown in figure 6.9. The Huber function is convex and first order differentiable.

The problem to solve in order to obtain the Huber-estimate is

$$\text{minimize} \sum_{i=1}^{m} \rho((Ax - b)_i). \tag{6.16}$$

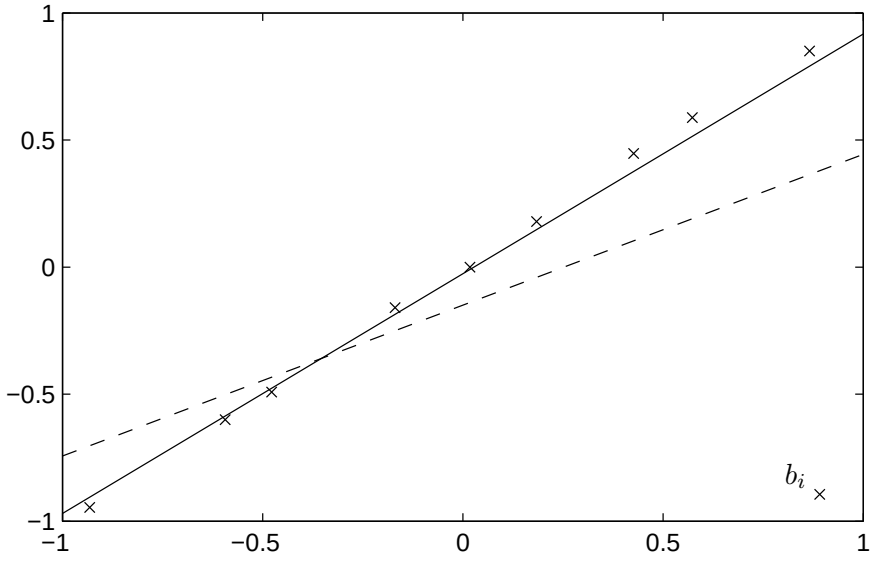


**Figure 6.8:** Huber M-estimate line fit to data that contains an outlier. The Least squares line fit is shown as the dashed line.

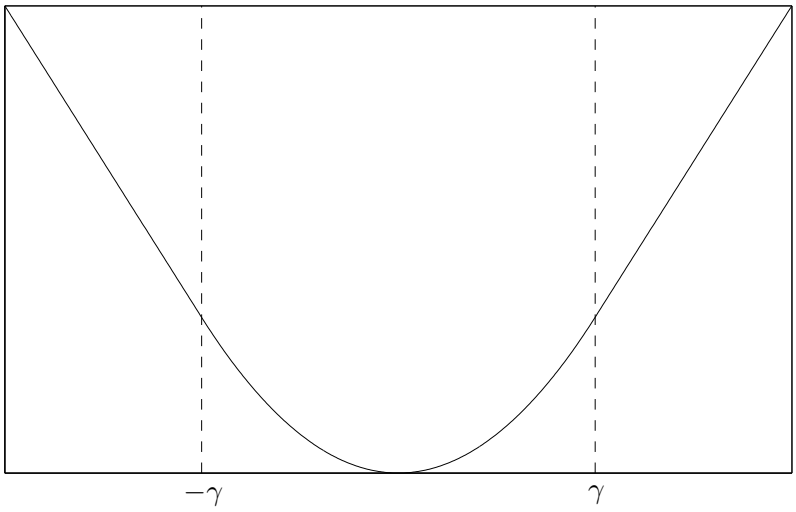


**Figure 6.9:** Huber M-estimator function.

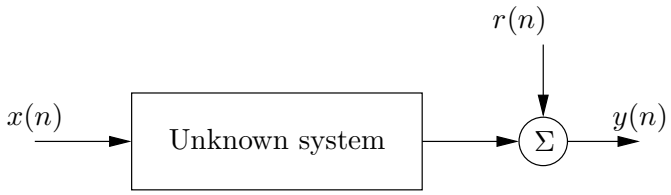


**Figure 6.10:** Unknown system identification subject to shot noise.

As shown by [Fuc99] this problem can be rewritten as

$$\text{minimize} \quad \tfrac{1}{2}\left\|b - Ax - z\right\|_2^2 + \gamma \left\|z\right\|_1 \tag{6.17}$$

where $z \in \mathbf{R}^m$ has been introduced. The optimal $x$ of (6.17) is also the optimal $x$ of (6.16). A similar result is presented in [MM00]. Under the assumption that $\mathbf{Rank}(A) = n$ the solution to (6.16) and therefore also (6.17) are unique.

Figure (6.8) show how the Huber M-estimator function can improve the line fit. Here $\gamma = 0.3$ which mean that data further than 0.3 units away from the line is considered to be an outlier.

### 6.5.3 A numerical example

#### Problem setup

This example consider the problem of identifying the impulse response of a linear system subject to shot noise. Given are a signal $x(n)$ and the output $y(n)$ of a system on the form

$$y(n) = \sum_{k=0}^{N} h(k)x(n-k) + r(n)$$

where $r(n)$ is the noise term. Such a system is shown in figure 6.10.

In the case where $r(n)$ is white Gaussian noise a good estimate of the impulse response is the least squares estimate, *i.e.* we an estimate $\hat{h}(n)$ is chosen such that it minimizes

$$\sum_n e(n)^2$$

where

$$e(n) = y(n) - \sum_{k=0}^{N} \hat{h}(k)x(n-k).$$

This example show how to estimate the impulse response in the case were the noise, $r(n)$, consist of a mixture of low variance white Gaussian noise and so called shot noise. The noise model is

$$r(n) = r_G(n) + r_I(n)$$

where $r_G(n)$ is small variance white Gaussian noise. The shot noise term $r_I(n)$ is modeled by

$$r_I(n) = \theta\eta$$

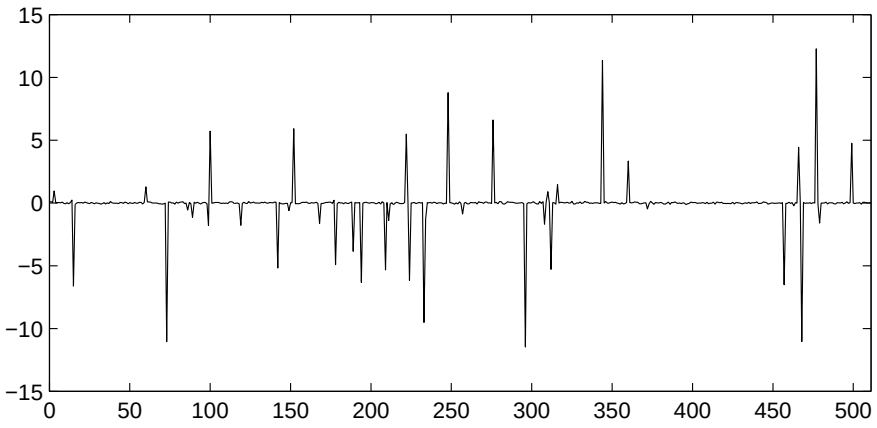


**Figure 6.11:** Additive shot noise r(n) composed of small variance Gaussian noise and shot noise.

| n | Original response | Least squares estimate | Huber M-estimate |
|---|---|---|---|
| 0 | 0.0007 | -0.0194 | 0.0057 |
| 1 | -0.0405 | -0.0707 | -0.0434 |
| 2 | -0.0450 | 0.0373 | -0.0449 |
| 3 | 0.0242 | 0.1301 | 0.0248 |
| 4 | 0.0731 | -0.0489 | 0.0680 |
| 5 | 0.0242 | -0.0845 | 0.0208 |
| 6 | -0.0450 | -0.0899 | -0.0431 |
| 7 | -0.0405 | 0.0698 | -0.0395 |
| 8 | 0.0007 | -0.0230 | -0.0013 |

**Table 6.2:** Impulse responses.

where $\theta$ is $0, 1$ binomially distributed with a probability $p$ and $\eta$ is drawn from a Laplace or a double sided exponential distribution with zero mean and variance $\lambda^2$. In the example $r_G(n)$ has variance $\sigma^2$ where $\sigma = 0.05$. The shot noise term has $p = 0.1$ and $\lambda = 4$, that is with 10% probability there will be a large Laplacian distributed term added to the Gaussian noise. The added noise is shown in figure 6.11.

In the problem setup 512 samples of the output signal of the system and the corresponding input signal are given. The input signal, for simplicity, is just ordinary Gaussian noise of unit variance. The true impulse response is a 9 tap FIR bandpass filter with linear phase. The impulse response is shown in figure 6.12 and table 6.2.

Due to the large components of the noise the least squares estimate of the impulse response does not give a satisfactory result. This in spite of the fact that roughly 90% of the samples are relatively clear.

The impulse response of the system is estimated using the Huber M-estimator with the parameter $\gamma = 0.1$. This correspond to 2 standard deviations of the Gaussian noise term.

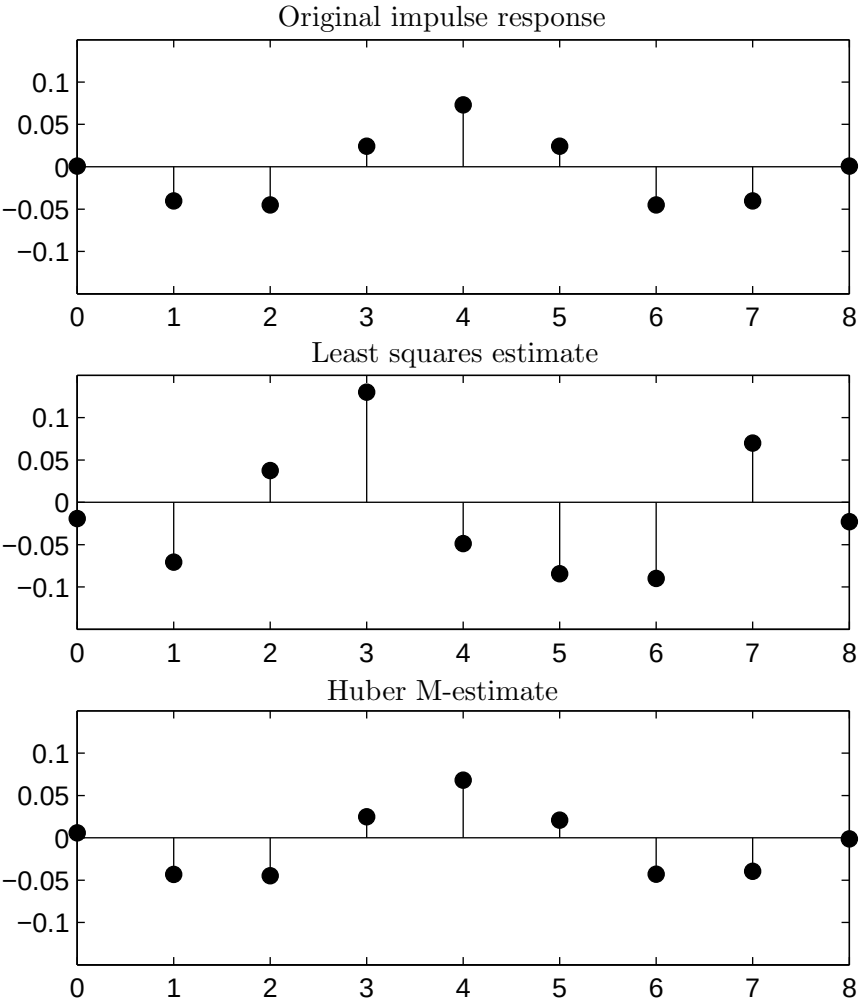


**Figure 6.12:** Impulse response as obtained by least squares estimate and Huber M-estimate. True response is shown as reference.

### Solution methodology

A dense 512 by 9 matrix $X$ is constructed as

$$X = \begin{bmatrix} x(8) & \dots & x(0) \\ \vdots & & \vdots \\ x(519) & \dots & x(511) \end{bmatrix}.$$

The output $y \in \mathbf{R}^{512}$ can be written as

$$y = Xh + r$$

where $h \in \mathbf{R}^9$ is the impulse response and $r \in \mathbf{R}^{512}$ is the noise.

The Huber M-estimate of the impulse response is obtained by solving

$$\text{minimize} \quad \left\| y - \begin{bmatrix} I & X \end{bmatrix} \begin{bmatrix} z \\ h \end{bmatrix} \right\|_2^2 + \gamma \left\| \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} z \\ h \end{bmatrix} \right\|_1 \tag{6.18}$$

over $z$ and $h$. Even though the added variable $z$ is of much higher dimension than the original variable $h$ the computational complexity does not grow significantly. The reason for this is that the matrix system matrix needed to be factored at each step of the algorithm has the form

$$H = \begin{bmatrix} 2I & X \\ X^T & X^T X \end{bmatrix}$$

and the Cholesky factorization of this matrix will be done efficiently.

The problem is solved using `l1l2optw` and converge to the specified tolerance of $10^{-8}$ in 9 iterations. The code runs in less than 0.3 seconds.

### Results

The result of the Huber M-estimate is shown along with the result of the least squares estimate in figure 6.12 and table 6.2. This example clearly show the ability of the Huber function to produce a satisfactory results in the presence of outliers.

As explained in [Fuc99] the added variable $z$ acts as an estimate of the shot noise term. The optimal $z$ is shown in figure 6.13.

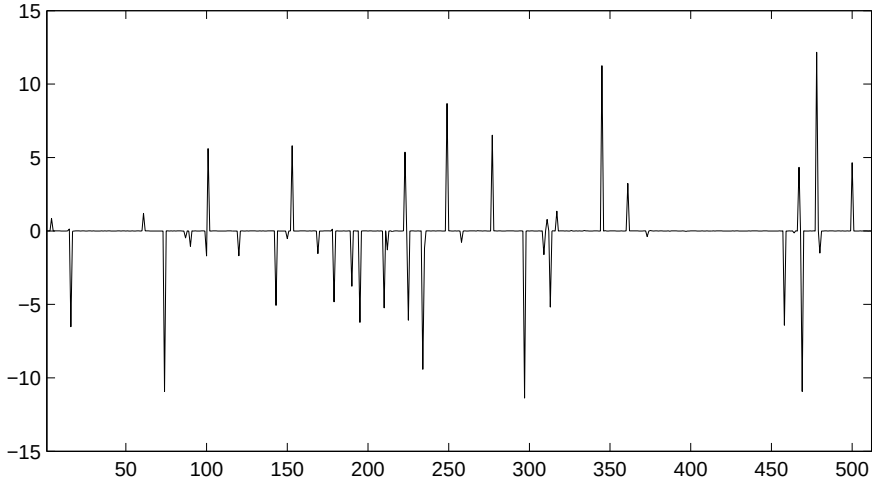


**Figure 6.13:** Optimal $z$ obtained from the solution of the Huber M-estimate. The variable $z$ can be viewed as an estimate of the shot noise term.

## 6.6 Optimal control

### 6.6.1 Linear quadratic regulators

A discrete time dynamical system of the form

$$\begin{aligned} &x(t+1) = Ax(t) + Bu(t), \qquad t = 0, \ldots, N-1 \\ &x(0) = x_0 \end{aligned} \tag{6.19}$$

is considered where $x(t)$ is some time dependent state variables and $u(t)$ are inputs to the system. The object will be to drive the states to zero from some given initial state by choosing appropriate input signals to the system. It will therefore be required that

$$x(N) = 0.$$

In a *linear quadratic regulator* (LQR) setting one introduce a quadratic cost function as a measure of the desirability of a certain solution to the above problem. This function, $J(x, u)$, is defined by

$$J(x,u) = \frac{1}{2}\sum_{t=1}^{N-1} x(t)^T P x(t) + \frac{1}{2}\sum_{t=0}^{N-1} u(t)^T Q u(t) \tag{6.20}$$

where $Q \succ 0$ and $P \succeq 0$. The desired solution is obtained as the solution of

$$\begin{array}{ll} \text{minimize} & J(x,u) \\ \text{subject to} & x(t+1) = Ax(t) + Bu(t), \qquad t = 0, \ldots, N-1 \\ & x(0) = x_0 \\ & x(N) = 0. \end{array} \tag{6.21}$$

The field of linear quadratic regulators has been extensively studied and there are several efficient ways of obtaining the solution to the above problem, for example dynamic programming.

### 6.6.2 Tradeoff between different objectives

New characteristics in the solution can be introduced by trading off some of the LQR objective $J(x, u)$ for some other objective. One such objective could for instance be

$$R_2(u) = \sum_{t=0}^{N-2} \|u(t+1) - u(t)\|_2^2 \tag{6.22}$$

which tend to produce smoother input signals $u(t)$. The new system would have the form

$$\begin{array}{ll} \text{minimize} & J(x,u) + \gamma R_2(u) \\ \text{subject to} & x(t+1) = Ax(t) + Bu(t), \qquad t = 0, \ldots, N-1 \\ & x(0) = x_0 \\ & x(N) = 0 \end{array} \tag{6.23}$$

where $\gamma \geq 0$ parameterize the tradeoff between the two objectives. This is similar to the quadratic smoothing regularization presented in section 6.4

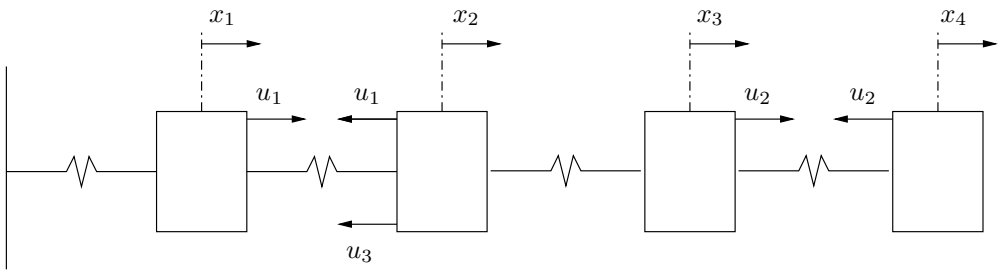


**Figure 6.14:** Simple mass spring system.

### 6.6.3 Piecewise constant controls

An interesting new objective is

$$R_1(u) = \sum_{t=0}^{N-2} \|u(t+1) - u(t)\|_1. \tag{6.24}$$

This objective function penalize the solution for having large variation in the input signals. By solving the problem

$$\begin{array}{ll} \text{minimize} & J(x,u) + \gamma R_1(u) \\ \text{subject to} & x(t+1) = Ax(t) + Bu(t), \qquad t = 0, \ldots, N-1 \\ & x(0) = x_0 \\ & x(N) = 0 \end{array} \tag{6.25}$$

Choosing a $\gamma > 0$ should be able to obtain piecewise constant controls as solutions of 6.25.

### 6.6.4 A numerical example

#### Problem setup

In the example piecewise constant controls for a system with dynamics given by

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.1 \\ -0.2 & 0.1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0.1 & -0.2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0.1 & -0.2 & 0.1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.1 & -0.1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.1 & 0 & 0 \\ -0.1 & 0 & -0.1 \\ 0 & 0.1 & 0 \\ 0 & -0.1 & 0 \end{bmatrix}.$$

will be obtained. This system represents a discretization of a simple one-dimensional mass-spring system with second order dynamics. This system in question is shown in figure 6.14. The states $x_1, \ldots, x_4$ are the offsets of the masses 1 to 4 from their equilibrium and the the states $x_5, \ldots, x_8$ are the current velocities of these masses. For simplicity all masses and spring constants are assumed to be 1. The input signals are forces applied as is illustrated in figure 6.14.

The object is thus to, from some given initial condition $x(0)$, steer the states of this system to zero in $N$ steps. Problem (6.25) will be solved to obtain a piecewise linear sequence of control inputs.

In this specific example $N = 60$. Simple $P$ and $Q$ matrices will also be used. These are $P = I$ and $Q = 10^{-2}I$. Due to the rank assumptions of our numeric solver the matrix $Q$ must be strictly positive definite but a small choice of $Q$ means that more emphasis is placed on the states, instead of the input signals, being close to zero.

**Solution methodology**

In order to solve (6.25) we rewrite this system as one large system. We arrange $x(t)$ for $t = 1, \ldots, N-1$ and $u(t)$ for $t = 0, \ldots, N-1$ in a vector $z$ as

$$z^T = \begin{bmatrix} u^T(0) & x^T(1) & u^T(1) & \ldots & x^T(N-1) & u^T(N-1) \end{bmatrix}.$$

$x(0)$ and $x(N)$ are excluded since these are given by directly by the constraints. The quadratic part of the objective function can be written as

$$J(z) = \frac{1}{2} \|Cz\|_2^2$$

where

$$C = \begin{bmatrix} Q^{1/2} & 0 & \cdots & 0 \\ 0 & P^{1/2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & Q^{1/2} \end{bmatrix}.$$

In this example $C$ is a 652 by 652 matrix. Similarly the $\ell_1$ part of the objective is expressed as

$$R_1(z) = \|Dz\|_1$$

where $D$ is the 180 by 652 matrix given by

$$D = \begin{bmatrix} I & 0 & -I & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & -I \end{bmatrix}.$$

The linear constrains are put into one system of the form

$$Fz = g$$

where $F$ is the 480 by 652 matrix given by

$$F = \begin{bmatrix} B & -I & 0 & 0 & \cdots & 0 & 0 \\ 0 & A & B & -I & & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & A & B \end{bmatrix}$$

and $g$ is given by

$$g = \begin{bmatrix} -Ax(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Problem (6.25) is thus rewritten as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Cz\|_2^2 + \gamma \|Dz\|_1 \\ \text{subject to} & Fz = g \end{array} \tag{6.26}$$

where $z \in \mathbf{R}^{652}$ is our variable of optimization. Due to the considerable sparsity of the system matrices the problem is quickly solved. For a value of $\gamma = 10$ the `MATLAB` function `l1l2optw` converge to a relative tolerance of $10^{-8}$ in 10 iterations. This takes approximately 1.1 seconds.

**Results**

Figure 6.15 show the input signals corresponding to the solution based of problem (6.21). In figure 6.16 the effect of introducing the $\ell_1$ objective is shown. The plot shown correspond to the solution of problem (6.25) for $\gamma = 10$.

As explained above piecewise constant inputs can be obtained by trading off some of the LQR objective for an objective that penalize input signals that are not constant in time.

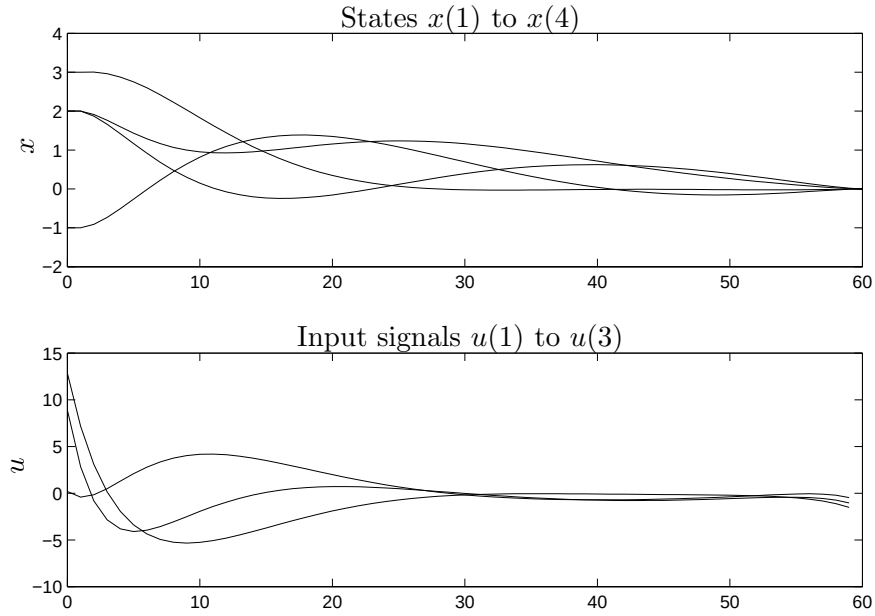


**Figure 6.15:** States and input signals as obtained for the original LQR objective.

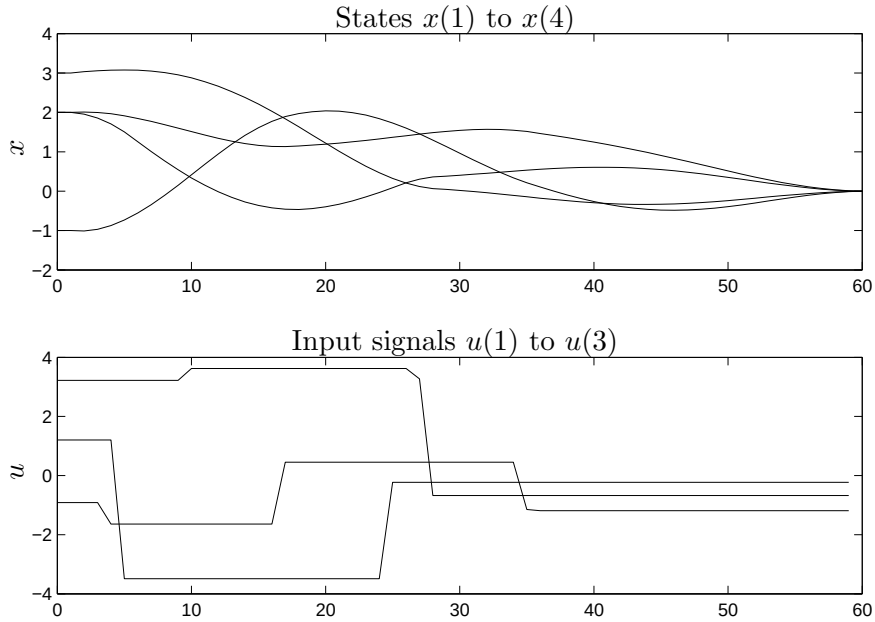


**Figure 6.16:** States and input signals as obtained when some of the LQR objective is traded off for an $\ell_1$ objective.

# Chapter 7

# Conclusions

This report has illustrated the usefulness of the $\ell_1$-norm as a second criterion in $\ell_2$-norm optimization problems. It has shown, both analytically and numerically through examples, that these bi-criterion problem can be solved with at a modest multiple of the time required to solve a similar $\ell_2$-norm problem.

Furthermore the report have investigated some mathematical properties of these problems and the solutions thereof. From the understanding of these problems an efficient interior point method based on the Mehrotra algorithm has been developed.

It has been the goal of this report to apart from mathematical analysis give an intuitive understanding of the types of applications that can be cast into this class of bi-criterion problems. The applications examined in the report are believed to be only a few examples of the potential applications. Since there is an abundance of examples of $\ell_2$-norm minimization there should be a large amount of these where the some of the optimality of the $\ell_2$-norm can be traded off for sparsity in the solution, perhaps the most striking example of this among those presented here is the piecewise constant controls. This should in particular hold true in applications where the $\ell_2$-norm is somewhat heuristically chosen as a criterion of optimality.

As the complexity, at least computationally, of these problems does not exceed a small multiple of similar $\ell_2$ problems the bi-criterion $\ell_1/\ell_2$-norm optimization should be viewed as a useful, and powerful, tool when it comes to solving many engineering problems.

# Appendix A

# l1l2optw and l1l2optc User's guide

`l1l2optw` and `l1l2optc` are two `MATLAB` functions that solve convex problems involving an $\ell_1$-norm (of an affine function) and a squared $\ell_2$-norm (of an affine function) subject to linear equality constraints. This is done by solving an associated quadratic program (QP) and its dual by a primal-dual interior-point method.

`l1l2optw` and `l1l2optc` are implementations of the algorithms presented in chapter 5.

## A.1 MATLAB function l1l2optw

### A.1.1 Problem definition

#### Primal and dual problems

The MATLAB function `l1l2optw` solves the (primal) problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2+\gamma\left\|Cx-d\right\|_1 \\ \text{subject to} & Fx=g \end{array} \tag{A.1}$$

and its associated dual problem

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\left\|\nu\right\|_2^2-b^T\nu-d^T\xi-g^T\chi \\ \text{subject to} & A^T\nu+C^T\xi+F^T\chi=0 \\ & \left\|\xi\right\|_\infty\leq\gamma. \end{array} \tag{A.2}$$

The primal and dual variables are

$$x\in\mathbf{R}^n, \qquad \xi\in\mathbf{R}^p, \qquad \nu\in\mathbf{R}^m, \qquad \chi\in\mathbf{R}^q,$$

and the problem parameters are

$$\gamma>0, \qquad A\in\mathbf{R}^{m\times n}, \qquad b\in\mathbf{R}^m, \qquad C\in\mathbf{R}^{p\times n}, \qquad d\in\mathbf{R}^p, \qquad F\in\mathbf{R}^{q\times n}, \qquad g\in\mathbf{R}^q \quad .$$

#### Optimality conditions

The optimality conditions for problem (A.1) and problem (A.2) are

$$\begin{aligned} Fx-g &= 0 && \text{(A.3a)} \\ A^T\nu+C^T\xi+F^T\chi &= 0 && \text{(A.3b)} \\ \left\|\xi\right\|_\infty &\leq \gamma && \text{(A.3c)} \\ Ax-b &= \nu && \text{(A.3d)} \\ \xi^T(Cx-d) &= \gamma\left\|Cx-d\right\|_1 && \text{(A.3e)} \end{aligned}$$

where (A.3a) is the primal constraint, (A.3b) and (A.3c) are the dual constraints and (A.3d) and (A.3e) are the complementary constraints.

$x^\star$ is primal optimal if and only if there are $\nu$, $\xi$ and $\chi$ such that equation (A.3) is satisfied. Similarly $\nu^\star$, $\xi^\star$ and $\chi^\star$ are dual optimal if and only if there is an $x$ such that (A.3) hold.

### A.1.2 Caveats

- $A$ and $C$ are assumed to have no common nullspace. This is the same as

$$\mathbf{Rank}\begin{bmatrix} A \\ C \end{bmatrix}=n.$$

This assumption is not due to the mathematics of the problem but it does imply that the matrix $A^TA + C^TC$ is positive definite which allows us to use the faster and more stable Cholesky factorization while computing search directions in the primal-dual interior-point code. The alternative would be using an $LDL^T$ factorization of a larger system to obtain the search direction. This might be included as an option in a later version.

- $F$ is assumed to be fat and full rank. That is $q < n$ and

  $$\mathbf{Rank}(F) = q.$$

  If $F$ does not satisfy this condition there are redundant or incompatible constraints which must be removed before $F$ is passed as a parameter to the code.

### A.1.3 Termination

Unless the code fails for some reason it will exit once it has obtained primal and dual variables satisfying

$$\begin{aligned}
r_{Fg} = \frac{\|Fx - g\|_2}{1 + \|g\|_2} &< \epsilon && \text{(A.4a)}\\
A^T\nu + C^T\xi + F^T\chi &= 0 && \text{(A.4b)}\\
\|\xi\|_\infty &\leq \gamma && \text{(A.4c)}\\
r_{Ab} = \frac{\|Ax - \nu - b\|_2}{1 + \|b\|_2} &< \epsilon && \text{(A.4d)}\\
r_{pd} = \frac{\gamma\|Cx - d\|_1 - \xi^T(Cx - d)}{1 + \gamma\|Cx - d\|_1} &< \epsilon. && \text{(A.4e)}
\end{aligned}$$

where $\epsilon > 0$ is some small tolerance. For specifying this tolerance refer to section (A.3).

Equation (A.4b) will be satisfied at each step of the iterative process and will thus be satisfied for the returned dual variables up to rounding errors due to finite precision arithmetics.

### A.1.4 Usage

```
[x,xi,nu] = l1l2optw(A,b,C,d,gamma)
```

solves problem (A.1) and problem (A.2) without equality constraints on $x$ and returns a primal optimal $x$ as well as dual optimal $\xi$ and $\nu$.

```
[x,xi,nu,chi] = l1l2optw(A,b,C,d,gamma,F,g)
```

solves problem problem (A.1) and problem (A.2) returns primal optimal $x$ as well as dual optimal $\xi$, $\nu$ and $\chi$.

```
[x,xi,nu,chi,iters] = l1l2optw(A,b,C,d,gamma,[],[],options)
[x,xi,nu,chi,iters] = l1l2optw(A,b,C,d,gamma,F,g,options)
```

is the full syntax of `l1l2optw` which returns the number of iterations required to converge to the solution and takes the optional parameter `options` overriding built in parameters.

`options` should be a `MATLAB` structure as described in section (A.3).

## A.2 MATLAB function l1l2optc

### A.2.1 Problem definition

#### Primal and dual problems

The MATLAB function `l1l2optc` solves the (primal) problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2 \\ \text{subject to} & \left\|Cx-d\right\|_1 \leq \alpha \\ & Fx = g \end{array} \tag{A.5}$$

and its associated dual problem

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\left\|\nu\right\|_2^2 - b^T\nu - d^T\xi - g^T\chi - \eta\alpha \\ \text{subject to} & A^T\nu + C^T\xi + F^T\chi = 0 \\ & \left\|\xi\right\|_\infty \leq \eta. \end{array} \tag{A.6}$$

The primal and dual variables are

$$x \in \mathbf{R}^n, \qquad \xi \in \mathbf{R}^p, \qquad \nu \in \mathbf{R}^m, \qquad \eta \in \mathbf{R}^+, \qquad \chi \in \mathbf{R}^q,$$

and the problem parameters are

$$\alpha \geq 0, \qquad A \in \mathbf{R}^{m\times n}, \qquad b \in \mathbf{R}^m, \qquad C \in \mathbf{R}^{p\times n}, \qquad d \in \mathbf{R}^p, \qquad F \in \mathbf{R}^{q\times n}, \qquad g \in \mathbf{R}^q \quad .$$

#### Optimality conditions

The optimality conditions for problem (A.5) and problem (A.6) are

$$\begin{aligned} Fx - g &= 0 && \text{(A.7a)} \\ A^T\nu + C^T\xi + F^T\chi &= 0 && \text{(A.7b)} \\ \left\|\xi\right\|_\infty &\leq \eta && \text{(A.7c)} \\ Ax - b &= \nu && \text{(A.7d)} \\ \xi^T(Cx-d) &= \eta\alpha. && \text{(A.7e)} \end{aligned}$$

where (A.7a) is the primal constraint, (A.7b) and (A.7c) are the dual constraints and (A.7d) and (A.7e) are the complementary constraints.

$x^\star$ is primal optimal if and only if there are $\nu$, $\xi$, $\eta$ and $\chi$ such that equation (A.7) is satisfied. Similarly $\nu^\star$, $\xi^\star$, $\eta^\star$ and $\chi^\star$ are dual optimal if and only if there is an $x$ such that (A.7) hold.

### A.2.2 Caveats

- $A$ and $C$ are assumed to have no common nullspace. This is the same as

$$\mathbf{Rank}\begin{bmatrix} A \\ C \end{bmatrix} = n.$$

- $F$ is assumed to be fat and full rank. That is $q < n$ and

$$\mathbf{Rank}(F) = q.$$

- $\alpha$ is assumed large enough for the problem to be feasible. No explicit check of feasibility will be performed and the code will simply reach the maximum number of iterations and give a warning if the problem is infeasible.

### A.2.3 Termination

Unless the code fails for some reason it will exit once it has obtained primal and dual variables satisfying

$$\begin{aligned} r_{Fg} = \frac{\|Fx - g\|_2}{1 + \|g\|_2} &< \epsilon && \text{(A.8a)} \\ A^T\nu + C^T\xi + F^T\chi &= 0 && \text{(A.8b)} \\ \|\xi\|_\infty &\leq \gamma && \text{(A.8c)} \\ r_{Ab} = \frac{\|Ax - \nu - b\|_2}{1 + \|b\|_2} &< \epsilon && \text{(A.8d)} \\ r_{pd} = \frac{|\eta\alpha - \xi^T(Cx - d)|}{1 + \eta\alpha} &< \epsilon. && \text{(A.8e)} \end{aligned}$$

where $\epsilon > 0$ is some small tolerance. For specifying this tolerance refer to section (A.3).

Equation (A.8b) will be satisfied at each step of the iterative process and will thus be satisfied for the returned dual variables up to rounding errors due to finite precision arithmetics.

### A.2.4 Usage

```
[x,xi,nu,eta] = l1l2optc(A,b,C,d,gamma)
```

solves problem (A.5) and problem (A.6) without the equality constraints on $x$ and returns primal optimal $x$ as well as dual optimal $\xi$, $\nu$ and $\eta$.

```
[x,xi,nu,eta,chi] = l1l2optc(A,b,C,d,gamma,F,g)
```

solves problem problem (A.5) and problem (A.6) returns primal optimal $x$ as well as dual optimal $\xi$, $\nu$, $\eta$ and $\chi$.

```
[x,xi,nu,eta,chi,iters] = l1l2optc(A,b,C,d,gamma,[],[],options)
[x,xi,nu,eta,chi,iters] = l1l2optc(A,b,C,d,gamma,F,g,options)
```

is the full syntax of `l1l2optc` which returns the number of iterations required to converge to the solution and takes the optional parameter `options` overriding built in parameters.

`options` should be a `MATLAB` structure as described in section (A.3).

## A.3 Options

### A.3.1 The options parameter

The `options` parameter passed to `l1l2optw` or `l1l2optc` should be a `MATLAB` structure containing one or more of the below specified fields.

- `options.maxiters` : Set the maximum number of iterations for the iterative process. If `options.maxiters` iterations has been reached before the specified tolerance the code will exit with a warning. The default value is 100.
- `options.tol` : Set the tolerance, $\epsilon$, of the solution. The code will exit once the specified tolerance is met. If the tolerance is set too small the code might fail due to matrices required for internal computations becoming poorly conditioned. The default value is 1e-8.
- `options.verbose` : The verbose option will make the code output progress reports to the terminal window. The verbose option let the user specify a level as follows
  - 0 : No output is made.
  - 1 : The quantities $r_{pd}$, $r_{Ab}$ and $r_{Fg}$ displayed at each iteration along with the number of iterations. For a specification of these quantities see section (A.1.3) and (A.2.3).

  A verbose option will output information of the specified level and of all lower levels. The default value is 0.

### A.3.2 Examples

```
x = l1l2optw(A,b,C,d,gamma,F,g,struct('maxiters',50,'tol',1e-6))
```

calls `l1l2optw` while restricting the number of iterations to 50 and setting the solution tolerance to $10^{-6}$.

```
options.maxiters = 50;
options.tol = 1e-6;
x = l1l2optw(A,b,C,d,gamma,F,g,options)
```

performs the exact same task as the above example.

```
x = l1l2optw(A,b,C,d,gamma,F,g,struct('verbose',1))
```

calls `l1l2optw` which will display information about the progress of the solution. This is especially helpful for large and computationally demanding problems.

## A.4 Technical details

### A.4.1 Efficiency

The code will produce a solution in a modest multiple of the number of operations required to produce the analytical solution of

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\left\|Ax-b\right\|_2^2+\gamma\frac{1}{2}\left\|Cx-d\right\|_2^2 \\ \text{subject to} & Fx=g. \end{array}$$

The modest multiple roughly corresponds to the number of iterations required to converge to the specified tolerance.

### A.4.2 Sparsity

Attempts will be made to perform the calculations using sparse matrix algebra. For large systems the computations will be efficient if

- `A`, `C` and `F` are all `MATLAB` sparse matrices.
- `A` and `C` are `MATLAB` sparse matrices and `F` is dense with $q \ll n$.

### A.4.3 Shortcomings

The code will fail (ungracefully) if any of the assumptions about the rank of the system matrices are not met, the matrices are extremely poorly conditioned or the requested tolerance of the solution is to small.

Parameter checking is performed to catch common mistakes. It is not however designed to be complete. Errors such as passing non-scalar values in the fields of the options parameter will lead to unpredictable results and possibly incomprehensible errors and warnings.

# Appendix B

# Notes on the MATLAB implementation

## B.1 Sparsity

In order to be able to handle problems of considerable size the implementation of the algorithms must be able to deal with sparse matrices. MATLAB provides an elegant framework for this through its sparse matrix data-types and functions.

An example is the total variation denoising application. At each iteration a system of equations of approximately a quarter of a million unknowns must be solved. Sparsity is what makes this possible at all.

### B.1.1 Matrix factorization

At each iteration we must solve a system of equations of the form

$$Hx = h \tag{B.1}$$

where

$$H = A^T A + C^T D^{-1} C \tag{B.2}$$

is a positive definite matrix. This is done by computing the Cholesky factorization of $H$ as

$$R^T R = H$$

where $R$ is a upper triangular matrix. The solution is then obtained by back substitution in two steps. That is, first $y = Rx$ is obtained as the solution of

$$R^T y = h$$

and then $x$ as the solution of

$$Rx = y.$$

The back substitutions are efficiently carried out by the MATLAB \ operator. Assuming that $H$ and $h$ are given system (B.1) can be solved as follows.

```
R = chol(H);    % Cholseky factorization
x = R\(R'\h');  % Back substitution
```

Since the majority of the operations are spent factorizing $H$ having obtained $R$ the system can be resolved for a different $h$ at a fraction of the cost of the first solution. This is used in the code to obtain both the affine scaling step and the correction and centering steps.

### B.1.2 Ordering

The ordering of the input variables can have a great impact on the sparsity pattern of the matrix $R$ of the Cholesky factorization [GL81]. In the code `l1l2optw` and `l1l2optc` symmetric minimum degree reordering is used. A permutation matrix $P$ is found such that

$$\bar{H} = PHP^T$$

has a sparser Cholesky factorization than H. In practice this is implemented as a reordering of the problem variables before the algorithm is started. In `MATLAB` this has the following form.

```
p = symmmd(A'*A+C'*C); % Symmetric minimum degree reordering
A = A(:,p)             % Reorder columns of A
C = C(:,p)             % Reorder columns of C
```

All other vectors and matrices dependent on the ordering of the variables must be permuted in a similar way. When returning the solution the variables are reordered to their original order.

## B.2 Numerical stability

### B.2.1 Stability of the factorizations

When close to the solution the matrix $D$ of (B.2) will be close to singular. This will case the matrix $H$ to be badly conditioned. Due to finite precision it might even become indefinite.

In an more stable implementation of the algorithm special care must be given to this problem. This is by no means a problem unique to this implementation but a rater common problem in interior point codes. In order to deal with this one could use a slight modification of the standard Cholesky algorithm, see [Wri99].

It is beyond the scope of this project to deal with more stable ways of computing this factorization other than the standard implementation of Cholesky factorization in `MATLAB`. Experiments show that special choices of the matrices $A$ and $C$ will cause the algorithm to fail to converge due to the conditioning of this system. In all applications considered here however the algorithm works well up the specified relative tolerance of $10^{-8}$.

### B.2.2 Alternative factorizations

As mentioned earlier when there are equality constrains a system of the form

$$\begin{bmatrix} H & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} h \\ g \end{bmatrix}.$$

must be solved. This system is indefinite and a good solution would be to solve this system using an $L^T DL$ factorization. Since this factorization is not currently available in `MATLAB` the above system is solved is solved using 2 Cholesky factorizations.

The same conclusions as stated above are true in this case as well. The chosen solution seems to be sufficient for the tested problems.

## B.3 Memory allocation

The main advantage of a C or Fortran implementation over an `MATLAB` implementation would likely be one of memory allocation. The sparsity pattern of the matrices involved in the algorithm does not change over time, this is something that could be exploited.

When for instance the system matrix $H$ of (B.1) is computed `MATLAB` is forced to reallocate a structure to store this Matrix. When faced with a complex sparsity pattern this could take considerable time. An C or Fortran implementation could precomputed this structure once and for all. The same is true for the sparsity pattern of the Cholesky factorization.

# Bibliography


[AR94] S. Alliney and S. Ruzinsky. An algorithm for the minimization of mixed $\ell_1$ and $\ell_2$ norms with application to bayesian estimation. *IEEE Transactions on Signal Processing*, 42(3):618–627, March 1994.

[BV01] S. Boyd and L. Vandenberghe. Introduction to convex optimization with engineering applications. Course Notes (available at http://www.stanford.edu/class/ee364). Stanford University, 2001.

[CDS99] S. S. Chen, D. L. D., and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[CGM99] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977, 1999.

[Dax91] A. Dax. A row relaxation method for large $\ell_1$ problems. *Linear algebra and its applications*, 154–156:793–818, 1991.

[Fuc99] J.-J. Fuchs. An inverse problem approach to robust regression. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4:1809–1812, March 1999.

[GL81] A. George and J. W.-H. Liu. *Computer solution of large sparse positive definite systems*. Prentice-Hall, 1981.

[GL96] G. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[Hub81] P. J. Huber. *Robust statistics*. Wiley, New York, 1981.

[LS96] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5:987–995, June 1996.

[Meh92] S. Mehrotra. On the implementation of a primal–dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[MM79] O. L. Mangasarian and R. R. Meyer. Nonlinear perturbation of linear programs. *SIAM Journal on Control and Optimization*, 17(6):745–, November 1979.

[MM00] O. L. Mangasarian and D. Musicant. Robust linear and support vector regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):950–955, September 2000.

[MS87] O. L. Mangasarian and T. H. Shiau. Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems. *SIAM Journal on Control and Optimization*, 25(3):583–595, 1987.

[OPT98] M. Osborne, B. Presnell, and B. Turlach. On the lasso and its dual. Technical report, Department of Statistics, University of Adelaide, 1998.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, Princeton, second edition, 1970.

[ROF92] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:241–252, 1992.

[SNT85] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*, volume 176 of *Mathematics in Science and Engineering*. Academic Press, 1985.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[TVW01] B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. Departement of Mathematics and Statistics, The University of Western Australia, March 2001.

[Vav94] Stephen A. Vavasis. Stable numerical algorithms for equilibrium systems. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1108–1131, 1994.

[Wil95] P. M. Williams. Bayesian regularization and pruning using a laplace prior. *Neural Computation*, 7(1):117–143, January 1995.

[Wri96] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, 1996.

[Wri99] S. J. Wright. Modified cholesky factorizations in interior-point algorithms for linear programming. *SIAM Journal on Optimization*, 9(4):1159–1191, 1999.