ROBUST AND CONVEX OPTIMIZATION WITH APPLICATIONS IN FINANCE

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Miguel Sousa Lobo
March 2000

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Stephen Boyd
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Darrell Duffie
(Graduate School of Business)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Benjamin Van Roy
(Engineering-Economic Systems Department)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Brad Osgood

Approved for the University Committee on Graduate Studies:

Books?
Books?
My god! You don't understand.
They were far too busy living first-hand
For books.
Books!

Joseph Moncure March, *The Wild Party* [42]

Acknowledgements

The work on second-order cone programming was done in collaboration with Lieven Vandenberghe and Stephen Boyd, and is indebted to the course notes for EE364 [15] (which should soon become a book.) Thanks also to Hervé Lebret, Michael Grant, Henry Wolkowicz, and Michael Todd for useful comments and suggestions.

Some of the work on portfolio optimization was co-authored with Maryam Fazel, in particular the relaxations for fixed transaction costs. The work on worst-case risk was done in close collaboration with Stephen Boyd. I am indebted for comments and suggestions from, among others, Marc Nunes, Darrell Duffie, Laurent El Ghaoui, Michael Saunders, Byunggyoo Kim, David Donoho, Bill Sharpe, John Watson, Stephen Wright, Garud Iyengar, Lieven Vandenberghe, and Haitham Hindi.

Most of my graduate studies were funded by a scholarship from the Portuguese Foundation for Science and Technology (ref. PRAXIS XXI / BD / 3625 / 94.) I also received some support from FLAD and AFOSR.

I would like to thank my committee, Darrell Duffie, Ben Van Roy, Brad Osgood, and Laurent El Ghaoui, as well as Thomas Kailath, who was instrumental in my admission to Stanford. A special thanks goes to my advisor, Stephen Boyd, who has been a pleasure to work with, and always a prolific source of ideas and support.

Finally, I am grateful for all the support I received from Monica, my family, and more friends than I could ever list.

Contents

List of Figures

Chapter 1

Introduction

1.1 Convex programs

Much of this work is motivated by relatively recent advances in nonlinear convex optimization. Our approach to a number of problems in portfolio optimization and worst-case risk analysis is made feasible by these developments, specifically by efficient interior-point methods that can handle problems with a large number of variables and constraints.

A convex program is an optimization problem where we seek the minimum of a convex function over a convex set. Among these, the most commonly used is the *linear program* (LP), an optimization problem with linear objective and linear inequality constraints:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, L, \end{array} \tag{1.1}$$

where the optimization variable is the vector x, and a_i, b_i, and c are problem parameters. Dantzig introduced the simplex method in 1948 [21], which led to the widespread use of linear programming.

While interior-point methods have been discussed for at least thirty years (see, *e.g.*, [27]), the current development was launched in 1984 by Karmarkar [31] with an

algorithm for linear programming that was more efficient than the simplex method. A large body of literature now exists on interior-point methods for linear programming, and a number of books have been written (*e.g.*, Wright [73] and Vanderbei [71]). Numerous implementations of efficient interior-point LP solvers are now available [70, 76, 20].

Of themselves, these developments did not change the traditional view that held nonlinear optimization problems to be fundamentally more difficult than linear ones. This would later be replaced with the understanding that the fundamental division in complexity lies in convex versus non-convex programming. Some ten years after Karmarkar presented his algorithm, Nesterov and Nemirovsky [50] noted that interior-point methods can be extended to handle many nonlinear convex optimization problems. Interior-point methods for nonlinear convex optimization problems have many of the same characteristics of the methods for linear programming. They have polynomial-time worst case complexity, and are extremely efficient in practice. Current algorithms can solve problems with hundreds of variables and constraints in times measured in seconds, or at most a few minutes, on a personal computer. If problem structure, such as sparsity, is exploited, much larger problems can be handled. The course notes by Boyd and Vandenberghe [15] give an accessible introduction to the field and describe a large number of applications.

A great amount of work has recently been done on some classes of nonlinear convex programs, both in terms of algorithms and applications. These include semidefinite programming (SDP) [68] and second-order cone programming (SOCP) [38]. A *second-order cone program* (SOCP) has the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \|A_i x + b_i\| \leq c_i^T x + d_i, \quad i = 1, \ldots, L, \end{array} \tag{1.2}$$

where $\|\cdot\|$ denotes the Euclidean norm, *i.e.*, $\|z\| = \sqrt{z^T z}$. SOCPs include linear and quadratic programming as special cases, but can also be used to solve a variety of nonlinear, nondifferentiable problems. Moreover, efficient interior-point software for SOCP is now available [37, 3, 63, 5].

In a *semi-definite program* (SDP) a matrix affine in the program variable x is constrained to be positive semi-definite:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & A_0 + A_1 x_1 + \ldots + A_n x_n \succeq 0, \end{array} \tag{1.3}$$

where $A_i \in \mathbf{R}^{n \times n}$, and $A \succeq 0$ denotes matrix inequality, *i.e.*, $z^T A z \geq 0, \forall z \in \mathbf{R}^n$. SDPs include LPs and SOCPs as special cases, but can also be used to solve many other nonlinear, nondifferentiable problems [68]. Efficient interior-point software for SDP is also available [67, 3, 63].

LP, SOCP, and SDP share important properties. All have a linear objective and a feasible set defined by the intersection of a hyperplane with a *self-dual convex cone*. They admit a *self-scaled barrier function*, and particularly efficient algorithms have been developed for these problems [52].

1.2 Robust optimization

In most practical optimization problems the problem data are uncertain or imprecise due, for instance, to estimation errors or to tolerances in design implementation. Linear programming (LP) is ubiquitously used in engineering, operations management, finance, and decision problems in many other fields where the problem data are prone to errors.

Recently, the idea of using robust optimization to deal with such parameter uncertainty has received some attention [9, 10, 26]. Robust optimization provides a novel and systematic approach for dealing with problem-data errors by, in effect, solving a min-max optimization problem. By treating the uncertainty in the data as deterministic, a solution is found which tolerates changes in problem data up to a certain error, and guarantees a certain level of performance in the face of uncertainty. Numerical evidence suggests that in many, if not most problems, a significant gain in the robustness of solution can be obtained at the expense of only a relatively small increase in the value of the program objective [10].

While no attempt is made here of a systematic review of robust optimization, the idea is present in much of this thesis. Chapter 2 includes the discussion of important examples of robust optimization problems that can be formulated as SOCPs: robust linear programming, and robust least-squares. Chapter 4 explores some applications of robust optimization in finance, and Chapter 5 discusses how sensitivity analysis of conic programs can be used in solution methods.

1.3 Motivation and overview

Many problems in finance can benefit greatly from new, efficient methods for large-scale convex optimization. Chapter 2 provides an overview of the theory and applications of SOCP which, among the convex optimization problems over self-dual cones, has been the last to be extensively studied. A number of portfolio optimization problems in the Markowitz mean-variance setting can be formulated as SOCPs, and hence efficiently solved.

In Chapter 3, we consider the problem of single-period portfolio selection, with transaction costs and constraints on exposure to risk. Linear transaction costs, bounds on the variance of the return, and bounds on different shortfall probabilities are shown to be efficiently handled by convex optimization methods. For such problems, the globally optimal portfolio can be computed very rapidly.

Portfolio optimization problems with transaction costs that include a fixed fee, or discount breakpoints, cannot be directly solved by convex optimization. For these problems, we describe a relaxation method which yields an easily computable upper bound via convex optimization. We also describe a heuristic method for finding a suboptimal portfolio, which is based on solving a small number of convex optimization problems (and hence can be done efficiently). Thus, we produce a suboptimal solution, and also an upper bound on the optimal solution. Numerical experiments suggest that for practical problems the gap between the two is small, even for large problems involving hundreds of assets. The same approach can be used for related problems, such as that of tracking an index with a portfolio consisting of a small number of assets.

Heuristics for nonconvex problem which involve the repeated solution of a convex program are made feasible due to the efficiency of solution methods for convex problems. Such heuristics and, even more so, convex relaxations, are an active avenue of research in many fields.

In Chapter 4, we show how to compute in a numerically efficient way the maximum risk of a portfolio, given uncertainty in the means and covariances of asset returns. This is an SDP, and is readily solved by interior-point methods. While not as general, this approach is more accurate and much faster than Monte Carlo methods. The computational effort required grows gracefully, so that very large problems can be handled. By using cutting-plane methods, the proposed approach is extended to portfolio selection, allowing "robust" portfolios to be designed. Note, however, that the ideas in this chapter are less thoroughly explored than those in the rest of this thesis, especially in regards to numerical experience.

Chapter 5 discusses some results on the perturbation of conic programs, which can be of use in solving robust optimization problems. Finally, Chapter 6 provides some brief final comments and directions for future work.

Chapter 2

Second-order cone programming

2.1 Introduction

Second-order cone programming is a problem class that lies between linear programming (or quadratic programming) and semidefinite programming. Like LP and SDP, SOCPs can be solved very efficiently by primal-dual interior-point methods (and in particular, far more efficiently than by treating the SOCP as an SDP). Moreover, a wide variety of practical problems can be formulated as second-order cone problems.

The main goal of this chapter is to present an overview of second-order cone programming. We start in §2.2 by describing several general convex optimization problems that can be cast as SOCPs. These problems include QP, QCQP, problems involving sums and maxima of norms, and hyperbolic constraints. We also describe two applications of SOCP to robust convex programming: robust LP and robust least-squares.

In Chapter 3 we will see practical applications in finance. In particular, several variations of portfolio optimization in the Markowitz framework, including problems with downside-risk constraints, can be solved via SOCP. Further, we describe a heuristic for portfolio problems with non-convex transaction costs that involves repeated solution of an SOCP. Such an approach is made viable because of the efficiency with which these convex programs can be solved.

In §2.3 we introduce the dual problem, and describe a primal-dual potential reduction method which is simple, robust, and efficient. The method we describe is certainly not the only possible choice: most of the interior-point methods that have been developed for linear (or semidefinite) programming can be generalized (or specialized) to handle SOCPs as well. The concepts underlying other primal-dual interior-point methods for SOCP, however, are very similar to the ideas behind the method presented here. An implementation of the algorithm (in C, with calls to LAPACK and BLAS, and with *cmex* interface to Matlab) is available via WWW or FTP [37]. The numerical examples in the next chapter were solved using this convex optimization software. Other software packages that handle this class of problems are now available, such as SDPPACK by Alizadeh *et al* [3], SeDuMi by Sturm [63], and MOSEK [5].

2.1.1 Second-order cone programming

We consider the *second-order cone program* (SOCP)

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\| \leq c_i^T x + d_i \quad i = 1, \ldots, N, \end{array} \tag{2.1}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and the problem parameters are $f \in \mathbf{R}^n$, $A_i \in \mathbf{R}^{(n_i-1)\times n}$, $b_i \in \mathbf{R}^{n_i-1}$, $c_i \in \mathbf{R}^n$, and $d_i \in \mathbf{R}$. The norm appearing in the constraints is the standard Euclidean norm, *i.e.*, $\|u\| = \left(u^T u\right)^{1/2}$. We call the constraint

$$\|A_i x + b_i\| \leq c_i^T x + d_i \tag{2.2}$$

a *second-order cone constraint of dimension* n_i, for the following reason. The standard or unit second-order (convex) cone of dimension k is defined as

$$\mathcal{C}_k = \left\{ \left[\begin{array}{c} u \\ t \end{array} \right] \;\middle|\; u \in \mathbf{R}^{k-1},\ t \in \mathbf{R},\ \|u\| \leq t \right\}$$

(which is also called the quadratic, ice-cream, or Lorentz cone). For $k = 1$ we define the unit second-order cone as

$$\mathcal{C}_1 = \{\ t \mid\ t \in \mathbf{R},\ 0 \leq t\ \}.$$

The set of points satisfying a second-order cone constraint is the inverse image of the unit second-order cone under an affine mapping:

$$\|A_i x + b_i\| \leq c_i^T x + d_i \iff \left[\begin{array}{c} A_i \\ c_i^T \end{array}\right] x + \left[\begin{array}{c} b_i \\ d_i \end{array}\right] \in \mathcal{C}_{n_i},$$

and hence is convex. Thus, the SOCP (2.1) is a convex programming problem since the objective is a convex function and the constraints define a convex set.

Second-order cone constraints can be used to represent several common convex constraints. For example, when $n_i = 1$ for $i = 1, \ldots, N$, the SOCP reduces to the linear program (LP)

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & 0 \leq c_i^T x + d_i \quad i = 1, \ldots, N. \end{array}$$

Another interesting special case arises when $c_i = 0$, so the ith second-order cone constraint reduces to $\|A_i x + b_i\| \leq d_i$, which is equivalent (assuming $d_i \geq 0$) to the (convex) quadratic constraint $\|A_i x + b_i\|^2 \leq d_i^2$. Thus, when all c_i vanish, the SOCP reduces to a quadratically constrained linear program (QCLP). We will soon see that (convex) quadratic programs (QPs), quadratically-constrained quadratic programs (QCQPs), and many other nonlinear convex optimization problems can be reformulated as SOCPs as well.

2.1.2 Related work

Much of the material in this chapter is covered in Lobo *et al.* [38]. This paper also describes a variety of engineering applications, including examples in filter design, antenna-array design, truss design, and robot-grasping-force optimization.

The main reference on interior-point methods for SOCP is the book by Nesterov and Nemirovsky [50]. The method we describe is the primal-dual algorithm of [50, §4.5] specialized to SOCP.

Adler and Alizadeh [1], Nemirovsky and Scheinberg [48], Tsuchiya [65] and Alizadeh and Schmieta [4] also discuss extensions of interior-point LP methods to SOCP. SOCP also fits the framework of optimization over *self-scaled* cones, for which Nesterov and Todd [51] have developed and analyzed a special class of primal-dual interior-point methods.

Other researchers have worked on interior-point methods for special cases of SOCP. One example is convex quadratic programming; see, for example, Den Hertog [23], Vanderbei [71], and Andersen and Andersen [6]. As another example, Andersen has developed an interior-point method for minimizing a sum of norms (which is a special case of SOCP; see §2.2.2), and describes extensive numerical tests in [7]. This problem is also studied by Xue and Ye [74] and Chan, Golub and Mulet [16]. Finally, Goldfarb, Liu and Wang [29] describe an interior-point method for convex quadratically constrained quadratic programming. Nesterov and Todd [52] provide a study of interior-point methods in the general framework of self-scaled cones (*i.e.*, LP, SOCP, and SDP.)

As noted in the introduction, a number of software packages that handle SOCP are now available [37, 3, 63, 5].

2.1.3 Relation to linear and semidefinite programing

We conclude this introduction with some general comments on the place of SOCP in convex optimization relative to other problem classes. SOCP includes several important standard classes of convex optimization problems, such as LP, QP and QCQP. On the other hand, it is itself less general than semidefinite programming (SDP), *i.e.*, the problem of minimizing a linear function over the intersection of an affine set and the cone of positive semidefinite matrices (see, *e.g.*, [68]). This can be seen as follows: The second-order cone can be embedded in the cone of positive

semidefinite matrices since

$$\|u\| \leq t \Longleftrightarrow \begin{bmatrix} tI & u \\ u^T & t \end{bmatrix} \succeq 0,$$

i.e., a second-order cone constraint is equivalent to a linear matrix inequality. (Here $\succeq$ denotes matrix inequality, *i.e.*, for $X = X^T \in \mathbf{R}^{n\times n}$, $X \succeq 0$ means $z^T X z \geq 0$ for all $z \in \mathbf{R}^n$.) Using this property the SOCP (2.1) can be expressed as an SDP

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \begin{bmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \quad i = 1, \ldots, N. \end{array} \tag{2.3}$$

Solving SOCPs via SDP is not a good idea, however. Interior-point methods that solve the SOCP directly have a much better worst-case complexity than an SDP method applied to (2.3): the number of iterations to decrease the duality gap to a constant fraction of itself is bounded above by $O(\sqrt{N})$ for the SOCP algorithm, and by $O(\sqrt{\sum_i n_i})$ for the SDP algorithm (see [50]). More importantly in practice, each iteration is much faster: the amount of work per iteration is $O(n^2 \sum_i n_i)$ in the SOCP algorithm and $O(n^2 \sum_i n_i^2)$ for the SDP. The difference between these numbers is significant if the dimensions n_i of the second-order constraints are large. A separate study of (and code for) SOCP is therefore warranted.

2.2 Problems that can be cast as SOCPs

In this section we describe some general classes of problems that can be formulated as SOCPs.

2.2.1 Quadratically constrained quadratic programming

We have already seen that an LP is readily expressed as an SOCP with 1-dimensional cones (*i.e.*, $n_i = 1$). Let us now consider the general *convex quadratically constrained*

quadratic program (QCQP)

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + 2q_0^T x + r_0 \\ \text{subject to} & x^T P_i x + 2q_i^T x + r_i \leq 0 \quad i = 1, \ldots, p, \end{array} \tag{2.4}$$

where $P_0,\ P_1,\ \ldots,\ P_p \in \mathbf{R}^{n\times n}$ are symmetric and positive semidefinite. We will assume for simplicity that the matrices P_i are positive definite, although the problem can be reduced to an SOCP in general. This allows us to write the QCQP (2.4) as

$$\begin{array}{ll} \text{minimize} & \left\|P_0^{1/2}x + P_0^{-1/2}q_0\right\|^2 + r_0 - q_0^T P_0^{-1} q_0 \\ \text{subject to} & \left\|P_i^{1/2}x + P_i^{-1/2}q_i\right\|^2 + r_i - q_i^T P_i^{-1} q_i \leq 0, \quad i = 1, \ldots, p, \end{array}$$

which can be solved via the SOCP with $p + 1$ constraints of dimension $n + 1$

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|P_0^{1/2}x + P_0^{-1/2}q_0\| \leq t, \\ & \|P_i^{1/2}x + P_i^{-1/2}q_i\| \leq \left(q_i^T P_i^{-1} q_i - r_i\right)^{1/2}, \quad i = 1, \ldots, p, \end{array} \tag{2.5}$$

where $t \in \mathbf{R}$ is a new optimization variable. The optimal values of (2.4) and (2.5) are equal up to a constant and a square root. More precisely, the optimal value of (2.4) is equal to $p^{*2} + r_0 - q_0^T P_0^{-1} q_0$, where p^* is the optimal value of (2.5).

As a special case, we can solve a *convex quadratic programming problem* (QP)

$$\begin{array}{ll} \text{minimize} & x^T P_0 x + 2q_0^T x + r_0 \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, p, \end{array}$$

($P_0 \succ 0$) as an SOCP with one constraint of dimension $n + 1$ and p constraints of dimension one:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|P_0^{1/2}x + P_0^{-1/2}q_0\| \leq t \\ & a_i^T x \leq b_i, \quad i = 1, \ldots, p, \end{array}$$

where the variables are x and t.

2.2.2 Problems involving sums and maxima of norms

Problems involving sums of norms are readily cast as SOCPs. Let $F_i \in \mathbf{R}^{n_i \times n}$ and $g_i \in \mathbf{R}^{n_i}$, $i = 1, \ldots, p$, be given. The unconstrained problem

$$\text{minimize} \quad \sum_{i=1}^{p} \|F_i x + g_i\|$$

can be expressed as an SOCP by introducing auxiliary variables $t_1, \ldots, t_p$:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} t_i \\ \text{subject to} & \|F_j x + g_j\| \leq t_j, \ j = 1, \ldots, p. \end{array}$$

The variables in this problem are $x \in \mathbf{R}^n$ and $t_i \in \mathbf{R}$. We can easily incorporate other second-order cone constraints in the problem, *e.g.*, linear inequalities on x.

The problem of minimizing a sum of norms arises in heuristics for the Steiner-tree problem [24, 74], optimal-location problems [53], and in total-variation image restoration [16]. Specialized methods are discussed in [7, 8, 19, 24, 16].

Similarly, problems involving a maximum of norms can be expressed as SOCPs: the problem

$$\text{minimize} \quad \max_{i=1,\ldots,p} \|F_i x + g_i\|$$

is equivalent to the SOCP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|F_i x + g_i\| \leq t, \quad i = 1, \ldots, p, \end{array}$$

in the variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

As a special case of the sum-of-norms problem, consider the complex ℓ_1-norm approximation problem:

$$\text{minimize} \quad \|Ax - b\|_1$$

where $x \in \mathbf{C}^q$, $A \in \mathbf{C}^{p \times q}$, $b \in \mathbf{C}^p$, and the ℓ_1-norm on $\mathbf{C}^p$ is defined by $\|v\|_1 = \sum_{i=1}^{p} |v_i|$. This problem can be expressed as an SOCP with p constraints of dimension

three:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} t_i \\ \text{subject to} & \left\| \begin{bmatrix} \mathbf{Re}\, a_i^T & -\mathbf{Im}\, a_i^T \\ \mathbf{Im}\, a_i^T & \mathbf{Re}\, a_i^T \end{bmatrix} z + \begin{bmatrix} \mathbf{Re}\, b_i \\ \mathbf{Im}\, b_i \end{bmatrix} \right\| \leq t_i, \quad i = 1, \ldots, p \end{array}$$

in the variables $z = [\mathbf{Re}\, x^T \;\; \mathbf{Im}\, x^T]^T \in \mathbf{R}^{2q}$, and t_i. In a similar way the complex ℓ_∞ norm approximation problem can be formulated as a maximum-of-norms problem.

As an extension that includes as special cases both the maximum and sum of norms, consider the problem of minimizing the sum of the k largest norms $\|F_i x + g_i\|$, *i.e.*, the problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{k} y_{[i]} \\ \text{subject to} & \|F_i x + g_i\| = y_i, \quad i = 1, \ldots, p, \end{array} \tag{2.6}$$

where $y_{[1]}, \; y_{[2]}, \ldots, y_{[p]}$ are the numbers $y_1, \; y_2, \ldots y_p$ sorted in decreasing order. It can be shown that the objective function in (2.6) is convex and that the problem is equivalent to the SOCP

$$\begin{array}{ll} \text{minimize} & kt + \displaystyle\sum_{i=1}^{p} y_i \\ \text{subject to} & \|F_i x + g_i\| \leq t + y_i, \quad i = 1, \ldots, p \\ & y_i \geq 0, \quad i = 1, \ldots, p, \end{array}$$

where the variables are x, $y \in \mathbf{R}^p$, and t. (See, *e.g.*, [69] or [15] for further discussion.)

2.2.3 Problems with hyperbolic constraints

Another large class of convex problems can be cast as SOCPs using the following fact:

$$w^2 \leq xy, \; x \geq 0, \; y \geq 0 \Longleftrightarrow \left\| \begin{bmatrix} 2w \\ x - y \end{bmatrix} \right\| \leq x + y, \tag{2.7}$$

and, more generally, when w is a vector,

$$w^T w \leq xy,\ x \geq 0,\ y \geq 0 \Longleftrightarrow \left\| \begin{bmatrix} 2w \\ x - y \end{bmatrix} \right\| \leq x + y. \tag{2.8}$$

We refer to these constraints as *hyperbolic constraints*, since they describe half of a hyperboloid.

As a first application, consider the problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} 1/(a_i^T x + b_i) \\ \text{subject to} & a_i^T x + b_i > 0, \quad i = 1, \ldots, p \\ & c_i^T x + d_i \geq 0, \quad i = 1, \ldots, q, \end{array}$$

which is convex since $1/(a_i^T x + b_i)$ is convex for $a_i^T x + b_i > 0$. This is the problem of maximizing the harmonic mean of some (positive) affine functions of x, over a polytope. This problem can be cast as an SOCP as follows. We first introduce new variables t_i and write the problem as one with hyperbolic constraints:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} t_i \\ \text{subject to} & t_i(a_i^T x + b_i) \geq 1, \quad t_i \geq 0, \quad i = 1, \ldots, p \\ & c_i^T x + d_i \geq 0, \quad i = 1, \ldots, q. \end{array}$$

By (2.7), this can be cast as an SOCP in x and t:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} t_i \\ \text{subject to} & \left\| \begin{bmatrix} 2 \\ a_i^T x + b_i - t_i \end{bmatrix} \right\| \leq a_i^T x + b_i + t_i, \quad i = 1, \ldots, p \\ & c_i^T x + d_i \geq 0, \quad i = 1, \ldots, q. \end{array}$$

As an extension, the quadratic/linear fractional problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} \frac{\|F_i x + g_i\|^2}{a_i^T x + b_i} \\ \text{subject to} & a_i^T x + b_i > 0, \quad i = 1, \ldots, p, \end{array}$$

where $F_i \in \mathbf{R}^{q_i \times n}$, $g_i \in \mathbf{R}^{q_i}$, can be cast as an SOCP by first expressing it as

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} t_i \\ \text{subject to} & (F_i x + g_i)^T (F_i x + g_i) \leq t_i (a_i^T x + b_i), \quad i = 1, \ldots, p \\ & a_i^T x + b_i > 0, \quad i = 1, \ldots, p, \end{array}$$

and then applying (2.8).

As another example, consider the logarithmic Chebyshev approximation problem,

$$\text{minimize} \quad \max_i |\log(a_i^T x) - \log(b_i)|, \tag{2.9}$$

where $A = [a_1 \cdots a_p]^T \in \mathbf{R}^{p \times n}$, $b \in \mathbf{R}^p$. We assume $b > 0$, and interpret $\log(a_i^T x)$ as $-\infty$ when $a_i^T x \leq 0$. The purpose of (2.9) is to approximately solve an over-determined set of equations $Ax \approx b$, measuring the error by the maximum logarithmic deviation between the numbers $a_i^T x$ and b_i. To cast this problem as an SOCP, first note that

$$|\log(a_i^T x) - \log(b_i)| = \log \max(a_i^T x / b_i, b_i / a_i^T x)$$

(assuming $a_i^T x > 0$). The log-Chebyshev problem (2.9) is therefore equivalent to minimizing $\max_i \max(a_i^T x / b_i, b_i / a_i^T x)$, or:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & 1/t \leq a_i^T x / b_i \leq t, \quad i = 1, \ldots, p. \end{array}$$

This can be expressed as the SOCP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & a_i^T x/b_i \le t, \quad i = 1, \ldots, p \\ & \left\| \begin{bmatrix} 2 \\ t - a_i^T x/b_i \end{bmatrix} \right\| \le t + a_i^T x/b_i, \quad i = 1, \ldots, p. \end{array}$$

As a final illustration of the use of hyperbolic constraints, we consider the problem of maximizing the geometric mean (or just product) of nonnegative affine functions (from Nesterov and Nemirovsky [50, §6.2.3, p.227]):

$$\begin{array}{ll} \text{maximize} & \displaystyle\prod_{i=1}^{p} (a_i^T x + b_i)^{1/p} \\ \text{suject to} & a_i^T x + b_i \ge 0, \quad i = 1, \ldots, p. \end{array}$$

For simplicity, we consider the special case $p = 4$; the extension to other values of p is straightforward. We first reformulate the problem by introducing new variables t_1, t_2, and t_3, and by adding hyperbolic constraints:

$$\begin{array}{ll} \text{maximize} & t_3 \\ \text{subject to} & (a_1^T x + b_2)(a_2^T x + b_2) \ge t_1^2, \quad a_1^T x + b_2 \ge 0, \quad a_2^T x + b_2 \ge 0 \\ & (a_3^T x + b_3)(a_4^T x + b_4) \ge t_2^2, \quad a_3^T x + b_3 \ge 0, \quad a_4^T x + b_4 \ge 0 \\ & t_1 t_2 \ge t_3^2, \quad t_1 \ge 0, \quad t_2 \ge 0. \end{array}$$

Applying (2.7) yields an SOCP.

2.2.4 Matrix-fractional problems

The next class of problems are *matrix-fractional* optimization problems of the form

$$\begin{array}{ll} \text{minimize} & (Fx+g)^T \left(P_0 + x_1P_1 + \cdots + x_pP_p\right)^{-1} (Fx+g) \\ \text{subject to} & P_0 + x_1P_1 + \cdots + x_pP_p \succ 0 \\ & x \geq 0, \end{array} \tag{2.10}$$

where $P_i = P_i^T \in \mathbf{R}^{n\times n}$, $F \in \mathbf{R}^{n\times p}$ and $g \in \mathbf{R}^n$, and the problem variable is $x \in \mathbf{R}^p$. (Here $A \succ B$ denotes strict matrix inequality, *i.e.*, $A - B$ positive definite, and $\geq$ denotes componentwise vector inequality.)

We first note that it is possible to solve this problem as an SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} P(x) & Fx+g \\ (Fx+g)^T & t \end{array}\right] \succeq 0, \end{array}$$

where $P(x) = P_0 + x_1P_1 + \cdots + x_pP_p$. The equivalence is readily demonstrated by using Schur complements, and holds even when the matrices P_i are indefinite. In the special case where the P_i are positive semidefinite, we can reformulate the matrix-fractional optimization problem more efficiently as an SOCP, as shown by Nesterov and Nemirovsky [50, §6.2.3, p.227]. We assume for simplicity that the matrix P_0 is nonsingular (see [50] for the general derivation).

We claim that (2.10) is equivalent to the following optimization problem in t_0, $\ldots$, $t_n \in \mathbf{R}$, y_0, y_1, $\ldots$, $y_p \in \mathbf{R}^n$, and x:

$$\begin{array}{ll} \text{minimize} & t_0 + t_1 + \cdots + t_p \\ \text{subject to} & P_0^{1/2}y_0 + P_1^{1/2}y_1 + \cdots + P_p^{1/2}y_p = Fx + g \\ & \|y_0\|^2 \leq t_0 \\ & \|y_i\|^2 \leq t_i x_i, \quad i = 1, \ldots, p \\ & t_i,\ x_i \geq 0 \quad i = 1, \ldots, p, \end{array} \tag{2.11}$$

which can be cast as an SOCP using (2.8):

$$\begin{array}{ll} \text{minimize} & t_0 + t_1 + \cdots + t_p \\ \text{subject to} & P_0^{1/2} y_0 + \sum_{i=1}^{p} P_i^{1/2} y_i = Fx + g \\ & \left\| \begin{bmatrix} 2y_0 \\ t_0 - 1 \end{bmatrix} \right\| \leq t_0 + 1, \\ & \left\| \begin{bmatrix} 2y_i \\ t_i - x_i \end{bmatrix} \right\| \leq t_i + x_i, \ i = 1, \ldots, p. \end{array}$$

The equivalence between (2.10) and (2.11) can be seen as follows. We first eliminate the variables t_i and reduce problem (2.11) to

$$\begin{array}{ll} \text{minimize} & y_0^T y_0 + y_1^T y_1 / x_1 + \cdots + y_p^T y_p / x_p \\ \text{subject to} & P_0^{1/2} y_0 + P_1^{1/2} y_1 + \cdots + P_p^{1/2} y_p = Fx + g \\ & x \geq 0 \end{array}$$

(interpreting $0/0 = 0$). Since the only constraint on y_i is the equality constraint, we can optimize over y_i by introducing a Lagrange multiplier $\lambda \in \mathbf{R}^n$ for the equality constraint, which gives us y_i in terms of λ and x:

$$2y_0 = -P_0^{1/2}\lambda \ \text{ and } \ 2y_i = -x_i P_i^{1/2}\lambda, \ \ i = 1, \ldots, p.$$

Next we substitute these expressions for y_i and obtain a minimization problem in λ and x:

$$\begin{array}{ll} \text{minimize} & \frac{1}{4}\lambda^T \left(P_0 + x_1 P_1 + \cdots + x_p P_p\right) \lambda \\ \text{subject to} & (P_0 + x_1 P_1 + \cdots + x_p P_p)\lambda = -2(Fx + g) \\ & x \geq 0. \end{array}$$

Finally, eliminating λ yields the matrix-fractional problem (2.10).

2.2.5 SOC-representable functions

The above examples illustrate several techniques that can be used to determine whether a convex optimization problem can be cast as an SOCP. In this section we formalize these ideas with the concept of a *second-order cone representation* of a set or function, introduced by Nesterov and Nemirovsky [50, §6.2.3].

We say a convex set $C \subseteq \mathbf{R}^n$ is *second-order cone representable* (abbreviated SOC-representable) if it can be represented by a number of second-order cone constraints, possibly after introducing auxiliary variables, *i.e.*, there exist $A_i \in \mathbf{R}^{(n_i-1)\times(n+m)}$, $b_i \in \mathbf{R}^{n_i-1}$, $c_i \in \mathbf{R}^{n+m}$, $d_i \in \mathbf{R}$, such that

$$x \in C \iff \exists y \in \mathbf{R}^m \text{ s.t. } \left\| A_i \begin{bmatrix} x \\ y \end{bmatrix} + b_i \right\| \leq c_i^T \begin{bmatrix} x \\ y \end{bmatrix} + d_i, \quad i = 1, \ldots, N.$$

We say a function f is second-order cone representable if its epigraph $\{(x,t) \mid f(x) \leq t\}$ has a second-order cone representation. The practical consequence is that if f and C are SOC-representable, then the convex optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

can be cast as an SOCP and efficiently solved via interior-point methods.

We have already encountered several examples of SOC-representable functions and sets. SOC-representable functions and sets can also be combined in various ways to yield new SOC-representable functions and sets. For example, if C_1 an C_2 are SOC-representable, then it is straightforward to show that αC_1 ($\alpha \geq 0$), $C_1 \cap C_2$ and $C_1 + C_2$ are SOC-representable. If f_1 and f_2 are SOC-representable functions, then αf_1 ($\alpha \geq 0$), $f_1 + f_2$, and $\max\{f_1, f_2\}$ are SOC-representable.

As a less obvious example, if f_1, f_2 are concave with $f_1(x) \geq 0$, $f_2(x) \geq 0$, and $-f_1$ and $-f_2$ are SOC-representable, then $f_1 f_2$ is concave and $-f_1 f_2$ is SOC-representable.

In other words the problem of maximizing the product of f_1 and f_2,

$$\begin{array}{ll} \text{maximize} & f_1(x)f_2(x) \\ \text{subject to} & f_1(x) \geq 0,\ f_2(x) \geq 0, \end{array}$$

can be cast as an SOCP by first expressing it as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & t_1 t_2 \geq t \\ & f_1(x) \geq t_1,\ \ f_2(x) \geq t_2 \\ & t_1 \geq 0,\ t_2 \geq 0, \end{array}$$

and then using the SOC-representation of $-f_1$ and $-f_2$.

SOC-representable functions are closed under composition. Suppose the convex functions f_1 and f_2 are SOC-representable and f_1 is monotone nondecreasing, so the composition g given by $g(x) = f_1(f_2(x))$ is also convex. Then g is SOC-representable. To see this, note that the epigraph of g can be expressed as

$$\{(x,t)|g(x) \leq t\} = \{(x,t)|\exists s \in \mathbf{R} \text{ s.t. } f_1(s) \leq t,\ f_2(x) \leq s\}$$

and the conditions $f_1(s) \leq t,\ f_2(x) \leq s$ can both be represented via second-order constraints.

2.2.6 Robust linear programming

In this section and the next we show how SOCP can be used to solve some simple robust convex optimization problems, in which uncertainty in the data is explicitly accounted for.

We consider a linear program,

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m, \end{array}$$

in which there is some uncertainty or variation in the parameters c, a_i, b_i. To simplify

the exposition, we will assume that c and b_i are fixed, and that the a_i are known to lie in given ellipsoids:

$$a_i \in \mathcal{E}_i = \{\, \overline{a}_i + P_i u \mid \|u\| \leq 1 \},$$

where $P_i = P_i^T \succeq 0$. (If P_i is singular we obtain 'flat' ellipsoids, of dimension $\operatorname{rank}(P_i)$).

In a worst-case framework, we require that the constraints be satisfied for all possible values of the parameters a_i, which leads us to the *robust linear program*

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \text{ for all } a_i \in \mathcal{E}_i, \quad i = 1, \ldots, m. \end{array} \tag{2.12}$$

The robust linear constraint $a_i^T x \leq b_i$ for all $a_i \in \mathcal{E}_i$ can be expressed as

$$\max\{\, a_i^T x \mid a_i \in \mathcal{E}_i \,\} = \overline{a}_i^T x + \|P_i x\| \leq b_i,$$

which is evidently a second-order cone constraint. Hence the robust LP (2.12) can be expressed as the SOCP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x + \|P_i x\| \leq b_i, \quad i = 1, \ldots, m. \end{array}$$

Note that the additional norm terms act as 'regularization terms,' discouraging large x in directions with considerable uncertainty in the parameters a_i. Note that conversely, we can interpret a general SOCP with $b_i = 0$ as a robust LP.

The robust LP can also be considered in a statistical framework (Whittle [72, §8.4]). Here we suppose that the parameters a_i are independent Gaussian random vectors, with mean $\overline{a}_i$ and covariance Σ_i. We require that each constraint $a_i^T x \leq b_i$ should hold with a probability (confidence) exceeding η, where $\eta \geq 0.5$, *i.e.*,

$$\operatorname{Prob}(a_i^T x \leq b_i) \geq \eta. \tag{2.13}$$

We will show that this probability constraint can be expressed as an SOC constraint.

Letting $u = a_i^T x$, with σ denoting its variance, this constraint can be written as

$$\text{Prob}\left(\frac{u-\overline{u}}{\sqrt{\sigma}} \leq \frac{b_i-\overline{u}}{\sqrt{\sigma}}\right) \geq \eta.$$

Since $(u-\overline{u})/\sqrt{\sigma}$ is a zero mean unit variance Gaussian variable, the probability above is simply $\Phi((b_i-\overline{u})/\sqrt{\sigma})$, where

$$\Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{-t^2/2}\,dt$$

is the CDF of a zero mean unit variance Gaussian random variable. Thus the probability constraint (2.13) can be expressed as

$$\frac{b_i-\overline{u}}{\sqrt{\sigma}} \geq \Phi^{-1}(\eta),$$

or, equivalently,

$$\overline{u} + \Phi^{-1}(\eta)\sqrt{\sigma} \leq b_i.$$

From $\overline{u} = \overline{a}_i^T x$ and $\sigma = x^T \Sigma_i x$ we obtain

$$\overline{a}_i^T x + \Phi^{-1}(\eta)\|\Sigma_i^{1/2} x\| \leq b_i.$$

Now, provided $\eta \geq 1/2$ (*i.e.*, $\Phi^{-1}(\eta) \geq 0$), this constraint is a second-order cone constraint.

In summary, the problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \text{Prob}\,(a_i^T x \leq b_i) \geq \eta, \quad i = 1, \ldots, m \end{array}$$

can be expressed as the SOCP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \overline{a}_i^T x + \Phi^{-1}(\eta)\|\Sigma_i^{1/2} x\| \leq b_i, \quad i = 1, \ldots, m. \end{array}$$

We refer to Ben-Tal and Nemirovsky [9], and Oustry, El Ghaoui, and Lebret [26]

for further discussion of robustness in convex optimization. For control applications of robust LP, see Boyd, Crusius and Hansson [13].

2.2.7 Robust least-squares

Suppose we are given an overdetermined set of equations $Ax \approx b$, where $A \in \mathbf{R}^{m\times n}, b \in \mathbf{R}^m$ are subject to unknown but bounded errors δA and δb with $\|\delta A\| \leq \rho$, $\|\delta b\| \leq \xi$ (where the matrix norm is the spectral norm, or maximum singular value). We define the *robust least-squares solution* as the solution $\widehat{x} \in \mathbf{R}^n$ that minimizes the largest possible residual, *i.e.*, $\widehat{x}$ is the solution of

$$\text{minimize} \quad \max_{\|\delta A\|\leq\rho,\ \|\delta b\|\leq\xi} \|(A+\delta A)x - (b+\delta b)\|. \tag{2.14}$$

This is the *robust least-squares problem* introduced by El Ghaoui and Lebret [25] and by Chandrasekaran, Golub, Gu and Sayed [17, 18, 58]. The objective function in problem (2.14) can be written in a closed form, by noting that

$$\begin{aligned}
\max_{\|\delta A\|\leq\rho,\ \|\delta b\|\leq\xi} \|(A+\delta A)x - (b+\delta b)\| \ &= \\
&= \max_{\|\delta A\|\leq\rho,\ \|\delta b\|\leq\xi} \max_{\|y\|\leq 1} y^T(Ax-b) + y^T\delta A x - y^T\delta b \\
&= \max_{\|z\|\leq\rho} \max_{\|y\|\leq 1} y^T(Ax-b) + z^T x + \xi \\
&= \|Ax-b\| + \rho\|x\| + \xi.
\end{aligned}$$

Problem (2.14) is therefore equivalent to minimizing a sum of Euclidean norms:

$$\text{minimize } \|Ax-b\| + \rho\|x\| + \xi.$$

Although this problem can be solved as an SOCP, there is a simpler solution via the singular-value decomposition of A. The SOCP-formulation becomes useful as soon as we put additional constraints on x, *e.g.*, nonnegativity constraints.

A variation on this problem is to assume that the rows a_i of A are subject to

independent errors, but known to lie in a given ellipsoid: $a_i \in \mathcal{E}_i$, where

$$\mathcal{E}_i = \{\overline{a}_i + P_i u \mid \|u\| \leq 1\} \qquad (P_i = P_i^T > 0).$$

We obtain the robust least squares estimate x by minimizing the worst-case residual:

$$\text{minimize} \quad \max_{a_i \in \mathcal{E}_i} \left(\sum_{i=1}^n \left(a_i^T x - b_i\right)^2\right)^{1/2}. \tag{2.15}$$

We first work out the objective function in a closed form:

$$\begin{aligned}
\max_{\|u\|\leq 1} \left|\overline{a}_i^T x - b_i + u^T P_i x\right| &= \max_{\|u\|\leq 1} \max \left\{\overline{a}_i^T x - b_i + u^T P_i x, -\overline{a}_i^T x + b_i - u^T P_i x\right\} \\
&= \max \left\{\overline{a}_i^T x - b_i + \|P_i x\|, -\overline{a}_i^T x + b_i + \|P_i x\|\right\} \\
&= \left|\overline{a}_i^T x - b_i\right| + \|P_i x\|.
\end{aligned}$$

Hence, the robust least-squares problem (2.15) can be formulated as

$$\text{minimize} \quad \left(\sum_{i=1}^n \left(\left|\overline{a}_i^T x - b_i\right| + \|P_i x\|\right)^2\right)^{1/2}$$

which can be cast as the SOCP

$$\begin{array}{ll}
\text{minimize} & s \\
\text{subject to} & \|t\| \leq s \\
& \left|\overline{a}_i^T x - b_i\right| + \|P_i x\| \leq t_i, \quad i = 1, \ldots, n.
\end{array}$$

These two robust variations on the least squares problem can be extended to allow for uncertainty on b. For the first problem, suppose the errors δA and δb are bounded as $\|[\delta A \ \delta b]\| \leq \rho$. Using the same analysis as above it can be shown that

$$\max_{\|[\delta A \ \ \delta b]\|\leq \rho} \|(A + \delta A)x - (b + \delta b)\| = \|Ax - b\| + \rho \left\| \begin{bmatrix} x \\ 1 \end{bmatrix} \right\|.$$

The robust least-squares solution can therefore be found by solving

$$\text{minimize } \|Ax - b\| + \rho \left\| \begin{bmatrix} x \\ 1 \end{bmatrix} \right\|.$$

In the second problem, we can assume b_i is bounded by $b_i \in [\overline{b}_i - p_i, \overline{b}_i + p_i]$. A straightforward calculation yields

$$\text{minimize } \left(\sum_{i=1}^{n} \left(\left| \overline{a}_i^T x - \overline{b}_i \right| + \|P_i x\| + p_i \right)^2 \right)^{1/2},$$

which can be easily cast as an SOCP.

2.3 Primal-dual interior-point method

In this section we outline the duality theory for SOCP, and briefly describe an efficient method for solving SOCPs. The method is the primal-dual potential reduction method of Nesterov and Nemirovsky [50, §4.5] applied to SOCP. When specialized to LP, the algorithm reduces to a variation of Ye's potential reduction method [75].

To simplify notation in (2.1), we will often use

$$u_i = A_i x + b_i, \quad t_i = c_i^T x + d_i, \quad i = 1, \ldots, N,$$

so that we can rewrite the SOCP problem (2.1) in the form

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|u_i\| \le t_i, \quad i = 1, \ldots, N \\ & u_i = A_i x + b_i, \; t_i = c_i^T x + d_i, \quad i = 1, \ldots, N. \end{array} \tag{2.16}$$

2.3.1 The dual SOCP

The *dual* of the SOCP (2.1) is given by

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^{N}\left(b_i^T z_i + d_i w_i\right) \\ \text{subject to} & \sum_{i=1}^{N}\left(A_i^T z_i + c_i w_i\right) = f \\ & \|z_i\| \leq w_i,\ i = 1, \ldots, N. \end{array} \tag{2.17}$$

The dual optimization variables are the vectors $z_i \in \mathbf{R}^{n_i - 1}$, and $w \in \mathbf{R}^N$. We denote a set of z_i's, $i = 1, \ldots, N$, by z. The dual SOCP (2.17) is also a convex programming problem since the objective (which is maximized) is concave, and the constraints are convex. Indeed, it has the same form as the SOCP in the form (2.16). Alternatively, by eliminating the equality constraints we can recast the dual SOCP in the same form as the original SOCP (2.1).

We will refer to the original SOCP as the *primal SOCP* when we need to distinguish it from the dual. The primal SOCP (2.1) is called *feasible* if there exists a *primal feasible* x, *i.e.*, an x that satisfies all constraints in (2.1). It is called *strictly feasible* if there exists a strictly primal feasible x, *i.e.*, an x that satisfies the constraints with strict inequality. The vectors z and w are called *dual feasible* if they satisfy the constraints in (2.17) and *strictly dual feasible* if in addition they satisfy $\|z_i\| < w_i$, $i = 1, \ldots, N$. We say the dual SOCP (2.17) is (strictly) feasible if there exist (strictly) feasible z_i, w. The optimal value of the primal SOCP (2.1) will be denoted as p^*, with the convention that $p^* = +\infty$ if the problem is infeasible. The optimal value of the dual SOCP (2.17) will be denoted as d^*, with $d^* = -\infty$ if the dual problem is infeasible.

The basic facts about the dual problem are:

1. *(weak duality)* $p^* \geq d^*$;

2. *(strong duality)* if the primal *or* dual problem is strictly feasible, then $p^* = d^*$;

3. if the primal *and* dual problems are strictly feasible, then there exist primal

and dual feasible points that attain the (equal) optimal values.

We only prove the first of these three facts; for a proof of 2 and 3, see, *e.g.*, Nesterov and Nemirovsky [50, §4.2.2].

The difference between the primal and dual objectives is called the *duality gap* associated with x, z, w, and will be denoted by $\eta(x, z, w)$, or simply η:

$$\eta(x, z, w) = f^T x + \sum_{i=1}^{N} \left(b_i^T z_i + d_i w_i \right). \tag{2.18}$$

Weak duality corresponds to the fact that the duality gap is always nonnegative, for any feasible x, z, w. To see this, we observe that the duality gap associated with primal and dual feasible points x, z, w can be expressed as a sum of nonnegative terms, by writing it in the form

$$\eta(x, z, w) = \sum_{i=1}^{N} \left(z_i^T (A_i x + b_i) + w_i (c_i^T x + d_i) \right) = \sum_{i=1}^{N} \left(z_i^T u_i + w_i t_i \right). \tag{2.19}$$

Each term in the right-hand sum is nonnegative:

$$z_i^T u_i + w_i t_i \geq -\|z_i\| \|u_i\| + w_i t_i \geq 0.$$

The first inequality follows from the Cauchy-Schwarz inequality. The second inequality follows from the fact that $t_i \geq \|u_i\| \geq 0$ and $w_i \geq \|z_i\| \geq 0$. Therefore $\eta(x, z, w) \geq 0$ for any feasible x, z, w, and as an immediate consequence we have $p^* \geq d^*$, *i.e.*, weak duality.

We can also reformulate part 3 of the duality result (which we do not prove here) as follows: If the problem is strictly primal and dual feasible, then there exist primal and dual feasible points with zero duality gap. By examining each term in (2.19), we see that the duality gap is zero if and only if the following conditions are satisfied for each i:

$$\|u_i\| < t_i \implies w_i = \|z_i\| = 0, \tag{2.20}$$

$$\|z_i\| < w_i \implies t_i = \|u_i\| = 0, \tag{2.21}$$

$$\|z_i\| = w_i, \quad \|u_i\| = t_i \quad \Longrightarrow \quad w_i u_i = -t_i z_i. \tag{2.22}$$

These three conditions generalize the *complementary slackness* conditions between optimal primal and dual solutions in LP. They also yield a sufficient condition for optimality: a primal feasible point x is optimal if, for $u_i = A_i x + b_i$ and $t_i = c_i^T x + d_i$, there exist z, w, such that (2.20)–(2.22) hold. (The conditions are also necessary if the primal and dual problems are strictly feasible.)

2.3.2 Barrier for second-order cone

We define, for $u \in \mathbf{R}^{m-1}$, $t \in \mathbf{R}$,

$$\phi(u,t) = \begin{cases} -\log\left(t^2 - \|u\|^2\right) & \|u\| < t \\ \infty & \text{otherwise.} \end{cases} \tag{2.23}$$

The function ϕ is a *barrier function* for the second-order cone $\mathcal{C}_m$: $\phi(u,t)$ is finite if and only if $(u,t) \in \mathcal{C}_m$ (*i.e.*, $\|u\| < t$), and $\phi(u,t)$ converges to ∞ as (u,t) approaches the boundary of $\mathcal{C}_m$. It is also smooth and convex on the interior of the second-order order cone. Its first and second derivatives are given by

$$\nabla\phi(u,t) = \frac{2}{t^2 - u^T u} \begin{bmatrix} u \\ -t \end{bmatrix}$$

and

$$\nabla^2\phi(u,t) = \frac{2}{(t^2 - u^T u)^2} \begin{bmatrix} (t^2 - u^T u)I + 2uu^T & -2tu \\ -2tu^T & t^2 + u^T u \end{bmatrix}.$$

2.3.3 Primal-dual potential function

For strictly feasible (x, z, w), we define the *primal-dual potential function* as

$$\varphi(x,z,w) = (2N + \nu\sqrt{2N}) \log\eta + \sum_{i=1}^{N} (\phi(u_i, t_i) + \phi(z_i, w_i)) - 2N \log N, \tag{2.24}$$

where $\nu \geq 1$ is an algorithm parameter, and η is the duality gap (2.18) associated with (x, z, w). The most important property of the potential function is the inequality

$$\eta(x, z, w) \leq \exp\left(\varphi(x, z, w) \Big/ \nu\sqrt{2N}\right), \tag{2.25}$$

which holds for all strictly feasible x, z, w. Therefore, if the potential function is small, the duality gap must be small. In particular, if $\varphi \to -\infty$, then $\eta \to 0$ and (x, z, w) approaches optimality.

The inequality (2.25) can be easily verified by noting the fact that

$$\psi(x, z, w) \triangleq 2N \log \eta + \sum_{i=1}^{N} (\phi(u_i, t_i) + \phi(z_i, w_i)) - 2N \log N \geq 0 \tag{2.26}$$

for all strictly feasible x, z, w. This implies $\varphi(x, z, w) \geq \nu\sqrt{2N} \log(\eta(x, z, w))$, and hence (2.25).

2.3.4 Primal-dual potential reduction algorithm

In a primal-dual potential reduction method, we start with strictly primal and dual feasible x, z, w and update them in such a way that the potential function $\varphi(x, z, w)$ is reduced at each iteration by at least some guaranteed amount. There exist several variations of this idea. In this section we present one such variation, the primal-dual potential reduction algorithm of Nesterov and Nemirovsky [50, §4.5].

At each iteration of the Nesterov and Nemirovsky method, primal and dual search directions δx, δz, δw are computed by solving the set of linear equations

$$\begin{bmatrix} H^{-1} & \bar{A} \\ \bar{A}^T & 0 \end{bmatrix} \begin{bmatrix} \delta Z \\ \delta x \end{bmatrix} = \begin{bmatrix} -H^{-1}(\rho Z + g) \\ 0 \end{bmatrix} \tag{2.27}$$

in the variables δx, δZ, where ρ is equal to $\rho = (2N + \nu\sqrt{2N})/\eta$, and

$$H = \begin{bmatrix} \nabla^2\phi(u_1, t_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \nabla^2\phi(u_N, t_N) \end{bmatrix}, \quad g = \begin{bmatrix} \nabla\phi(u_1, t_1) \\ \vdots \\ \nabla\phi(u_N, t_N) \end{bmatrix},$$

$$Z = \begin{bmatrix} z_1^T & w_1 & \cdots & z_N^T & w_N \end{bmatrix}^T, \quad \delta Z = \begin{bmatrix} \delta z_1^T & \delta w_1 & \cdots & \delta z_N^T & \delta w_N \end{bmatrix}^T.$$

The outline of the algorithm is as follows.

Primal-dual potential reduction algorithm

given strictly feasible x, z, w, a tolerance $\epsilon > 0$, and a parameter $\nu \geq 1$.

repeat

1. Find primal and dual search directions by solving (2.27).
2. *Plane search.* Find p, $q \in \mathbf{R}$ that minimize $\varphi(x + p\delta x, z + q\delta z, w + q\delta w)$.
3. *Update.* $x := x + p\delta x$, $z := z + q\delta z$, $w := w + q\delta w$.

until $\eta(x, z, w) \leq \epsilon$.

It can be shown that at each iteration of the algorithm, the potential function decreases by at least a fixed amount:

$$\varphi(x^{(k+1)}, z^{(k+1)}, w^{(k+1)}) \leq \varphi(x^{(k)}, z^{(k)}, w^{(k)}) - \delta$$

where $\delta > 0$ does not depend on any problem data at all (including the dimensions). For a proof of this result, see [50, §4.5]. Combined with (2.25) this provides a bound on the number of iterations required to attain a given accuracy ϵ. From (2.25) we see that $\eta \leq \epsilon$ after at most

$$\frac{\nu\sqrt{2N}\log(\eta^{(0)}/\epsilon) + \psi(x^{(0)}, z^{(0)}, w^{(0)})}{\delta}$$

iterations. Roughly speaking and provided the initial value of ψ is small enough, this means it takes no more than $O(\sqrt{N})$ steps to reduce the initial duality gap by a given factor.

Computationally the most demanding step in the algorithm is solving the linear system (2.27). This can be done by first eliminating δZ from the first equation, solving

$$\bar{A}^T H \bar{A}\, \delta x = -\bar{A}^T(\rho Z + g) = -\rho f - \bar{A}^T g \tag{2.28}$$

for δx, and then substituting to find

$$\delta Z = -\rho Z - g - H\bar{A}\delta x.$$

Since $\bar{A}^T \delta Z = 0$, the updated dual point $z + q\delta z$, $w + q\delta w$ satisfies the dual equality constraints, for any $q \in \mathbf{R}$.

An alternative is to directly solve the larger system (2.27) instead of (2.28). This may be preferable when $\bar{A}$ is very large and sparse, or when the equations (2.28) are badly conditoned. Note that

$$\nabla^2\phi(u,t)^{-1} = \frac{1}{2}\begin{bmatrix} (t^2 - u^T u)I + 2uu^T & 2tu \\ 2tu^T & t^2 + u^T u \end{bmatrix},$$

and therefore forming $H^{-1} = \mathbf{diag}\,(\nabla^2\phi(u_1,t_1)^{-1}, \ldots, \nabla^2\phi(u_N,t_N)^{-1})$ does not require a matrix inversion.

We refer to the second step in the algorithm as the *plane search* since we are minimizing the potential function over the plane defined by the current points x,z,w and the current primal and dual search directions. This plane search can be carried out very efficiently using some preliminary preprocessing, similar to the plane search in potential reduction methods for SDP [68].

We conclude this section by pointing out the analogy between (2.27) and the systems of equations arising in interior-point methods for LP. We consider the primal-dual pair of LPs

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & c_i^T x + d_i \geq 0, \quad i = 1, \ldots, N \end{array}$$

and

$$\begin{array}{ll} \text{minimize} & -\sum_{i=1}^{N} d_i z_i \\ \text{subject to} & \sum_{i=1}^{N} z_i c_i = f \\ & z_i \geq 0, \quad i = 1, \ldots, N, \end{array}$$

and solve them as SOCPs with $n_i = 1$, $i = 1, \ldots, N$. Using the method outlined above, we obtain

$$\bar{A} = [c_1 \quad \cdots \quad c_N]^T, \quad \bar{b} = d,$$

and writing $X = \mathbf{diag}\left(c_1^T x + d_1, \ldots, c_N^T x + d_N\right)$, the equation (2.27) reduces to

$$\begin{bmatrix} \frac{1}{2}X^2 & \bar{A} \\ \bar{A}^T & 0 \end{bmatrix} \begin{bmatrix} \delta z \\ \delta x \end{bmatrix} = \begin{bmatrix} -(\rho/2)X^2 z + Xe \\ 0 \end{bmatrix}, \tag{2.29}$$

where e is the vector with all components equal to one. The factor $1/2$ in the first block can be absorbed into δz since only the direction of δz is important, and not its magnitude. Also note that $\rho/2 = (N + \nu\sqrt{N})/\eta$. We therefore see that the equations (2.29) coincide with (one particular variation) of familiar expressions from LP.

2.3.5 Finding strictly feasible initial points

The algorithm of the previous section requires strictly feasible primal and dual starting points. In this section we discuss two techniques that can be used when primal and/or dual feasible points are not readily available.

Bounds on the primal variables

It is usually easy to find strictly dual feasible points in SOCPs when the primal constraints include explicit bounds on the variables, *e.g.*, componentwise upper and lower bounds $l \leq x \leq u$, or a norm constraint $\|x\| \leq R$. For example, suppose that

we modify the SOCP (2.1) by adding a bound on the norm of x:

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\| \leq c_i^T x + d_i, \quad i = 1, \ldots, N \\ & \|x\| \leq R. \end{array} \tag{2.30}$$

If R is large enough, the extra constraint does not change the solution and the optimal value of the SOCP. The dual of the SOCP (2.30) is

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^{N} \left(b_i^T z_i + d_i w_i \right) - R w_{N+1} \\ \text{subject to} & \sum_{i=1}^{N} \left(A_i^T z_i + c_i w_i \right) + z_{N+1} = f \\ & \|z_i\| \leq w_i, \quad i = 1, \ldots, N+1. \end{array} \tag{2.31}$$

Strictly feasible points for (2.31) can be easily calculated as follows. For $i = 1, \ldots, N$, we can take any z_i and $w_i > \|z_i\|$. The variable z_{N+1} then follows from the equality constraint in (2.31), and for w_{N+1} we can take any number greater than $\|z_{N+1}\|$.

This idea of adding bounds on the primal variable is a variation on the big-M method in linear programming [73].

Phase-I method

A primal strictly feasible point can be computed by solving the SOCP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|A_i x + b_i\| \leq c_i^T x + d_i + t, \quad i = 1, \ldots, N, \end{array} \tag{2.32}$$

in the variables x and t. If (x, t) is feasible in (2.32), and $t < 0$, then x satisfies $\|A_i x + b_i\| < c_i^T x + d_i$, *i.e.*, it is strictly feasible for the original SOCP (2.1). We can therefore find a strictly feasible x by solving (2.32), provided the optimal value $t^\star$ of the SOCP (2.32) is negative. If $t^\star > 0$, the original SOCP (2.1) is infeasible.

Note that it is easy to find a strictly feasible point for the SOCP (2.32). One

possible choice is

$$x = 0, \quad t > \max_i \left(\|b_i\| - d_i \right).$$

The dual of the SOCP (2.32) is

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{N} \left(b_i^T z_i + d_i w_i \right) \\ \text{subject to} & \displaystyle\sum_{i=1}^{N} \left(A_i^T z_i + c_i w_i \right) = 0 \\ & \displaystyle\sum_{i=1}^{N} w_i = 1 \\ & \|z_i\| \leq w_i, \quad i = 1, \ldots, N. \end{array} \tag{2.33}$$

If a strictly feasible (z, w) for (2.33) is available, one can solve the phase-I problem by applying the primal-dual algorithm of the previous section to the pair of problems (2.32) and (2.33). If no strictly feasible (z, w) for (2.33) is available, one can add an explicit bound on the primal variable as described above.

2.3.6 Performance in practice

Our experience with the method is consistent with the practical behavior observed in many similar methods for linear or semidefinite programming: the number of iterations is only weakly dependent on the problem dimensions (n, n_i, N), and typically lies between 5 and 50 for a very wide range of problem sizes.

Thus we believe that for practical purposes the cost of solving an SOCP is roughly equal to the cost of solving a modest number (5–50) of systems of the form (2.28). If no special structure in the problem data is exploited, the cost of solving the system is $O(n^3)$, and the cost of forming the system matrix is $O(n^2 \sum_{i=1}^{N} n_i)$. In practice, special problem structure (*e.g.*, sparsity) often allows forming the equations faster, or solving the systems (2.27) or (2.28) more efficiently.

We close this section by pointing out a few possible improvements. The most popular interior-point methods for linear programming share many of the features of the potential reduction method we presented here, but differ in three respects (see [73]).

First, they treat the primal and dual problems more symmetrically (for example, the diagonal matrix X^2 in (2.29) is replaced by XZ^{-1}). A second difference is that common interior-point methods for LP are one-phase methods that allow an infeasible starting point. Finally, the asymptotic convergence of the method is improved by the use of predictor steps. These different techniques can all be extended to SOCP. In particular, Nesterov and Todd [51], Alizadeh *et al.* [1, 4, 2], and Tsuchiya [65] have recently developed extensions of the symmetric primal-dual LP methods to SOCP.

Chapter 3

Portfolio optimization

3.1 Introduction

This chapter deals with the problem of single-period portfolio optimization. We consider the maximization of expected return, taking transaction costs into account, and subject to different types of constraints on the allowable portfolios.

Our approach is based on the fact that *convex* optimization problems, even if nonlinear or large-scale, can be numerically solved with great efficiency, using recently developed algorithms. We show that a number of portfolio optimization problems can be cast as convex optimization problems, and hence globally, and efficiently, solved. This class of convex portfolio optimization problems includes those with linear transactions costs, margin and diversity constraints, and limits on variance and on shortfall risk. In fact, all of these problems can be cast as SOCPs, and the numerical examples presented here were solved using the algorithm described in Chapter 2.

We also consider problems with fixed transaction costs (possibly in addition to linear transaction costs). These nonconvex portfolio optimization problems cannot be solved directly via convex optimization, but we describe two approaches that are based on convex optimization. These problems can be solved exactly (*i.e.*, globally) by solving a number of convex problems that, unfortunately, grows exponentially with the number of assets. This method, as well as other more sophisticated methods of global optimization, is practical only for portfolios with about 15 or fewer assets.

Our main contribution is to describe a method for *approximately* solving much larger nonconvex portfolio optimization problems, by solving a small number of convex optimization problems. The method yields a possibly suboptimal portfolio, as well as a guaranteed upper bound on the global optimum. While there is no guarantee that the gap between the performance of the suboptimal portfolio and the upper bound will be small, we find that in practice it is. Our method therefore gives an effective practical solution to nonconvex portfolio optimization problems, even with many hundreds of assets and portfolio constraints. If higher guaranteed accuracy is needed, our method can be embedded in a branch-and-bound algorithm.

The unifying idea in this chapter is to exploit new efficient interior-point methods for convex optimization. While such methods are of polynomial complexity (in simple implementations, cubic) in problem dimension, the availability of computing resources over time shows no signs of departing from geometric growth. As a consequence, interior-point methods will be able to handle very large problems, in very short run-times in the near future. Currently, run-times are on the order of a minute for problems with a few hundred variables, on an inexpensive personal computer, using generic software that is not optimized for portfolio problems.

The single-period portfolio selection problem is stated in §3.2. Transaction costs functions and portfolio constraints are described in §3.3 and §3.4. An example of a convex problem with linear transaction costs is presented in §3.5. Fixed costs are included in §3.6, where it is shown how to compute a global bound on performance and how to obtain an approximate solution. Numerical examples are given in §3.7. Related problems, such as index tracking, are briefly discussed in §3.8.

3.1.1 Related work

Broadly speaking, our approach falls in the Markowitz framework, where a tradeoff between return mean and variance is present. The genesis of the field has been independently attributed to Markowitz [43, 44] and Roy [56]. Implications for the valuation of assets arose with the capital asset pricing model (CAPM) of Sharpe [62] and Lintner [35]. Recent general references are, *e.g.*, Rudolf [57], and Luenberger [41].

The book from Salomon Brothers [34] is one of many sources for the downside-risk approach, which has been increasingly used in recent years (although it was already described in Roy's 1952 paper.)

For fixed transaction costs, solutions have been found for specific structures of the covariance matrix. Blog *et al.* [12], describe a solution for a single-factor model, *i.e.*, a diagonal plus rank-one covariance matrix. Patel and Subrahmanyam [54]) assume an even more specific structure, namely that there is an identical correlation coefficient between all assets and the single factor. In contrast, we make no assumptions about the correlation matrix, and moreover, allow the addition of any other (convex) cost terms and constraints.

Many treatments have been presented for problems with linear costs. Most methods described in the literature are modifications of the simplex method for quadratic programming, which can handle a quadratic objective but not quadratic constraints. The variance is included in the program objective, weighted by a parameter λ, and the solutions on the efficient frontier are found by varying the parameter λ. See, *e.g.*, Perold [55], where a method for efficiently ranging over such a parameterization of the efficient frontier is proposed.

The iterative heuristic we propose for finding a suboptimal solution was developed simultaneously and independently by Jason Schattman [59]. It is also related to one given by Delaney and Bresler [22], in the context of image reconstruction. Meyer [45] establishes the convergence of a large class of algorithms that includes the heuristic discussed in this chapter.

For branch-and-bound methods and integer programming, which, combined with the methods described in this chapter, can be used to solve the nonconvex portfolio optimization problems exactly, see, *e.g.*, Lawler and Wood [33], and Schrijver [61].

3.2 The portfolio selection problem

Consider an investment portfolio that consists of holdings in some or all of n assets. This portfolio is to be adjusted by performing a number of transactions, after which the portfolio will be held over a fixed time period. The investor's goal is to maximize

the expected wealth at the end of period, while satisfying a set of constraints on the portfolio. These constraints typically include limits on exposure to risk, and bounds on the amount held in each asset. The problem of an investor averse to risk in terms of "mean-variance" preferences can be treated in a similar fashion.

The current holdings in each asset are $w = (w_1, \ldots, w_n)$. The total current wealth is then $\mathbf{1}^T w$, where $\mathbf{1}$ is a vector with all entries equal to one. The dollar amount transacted in each asset is specified by $x = (x_1, \ldots, x_n)$, with $x_i > 0$ for buying, $x_i < 0$ for selling. After transactions, the adjusted portfolio is $w + x$. Representing the sum of all transaction costs associated with x by $\phi(x)$, the budget constraint is

$$\mathbf{1}^T x + \phi(x) = 0. \tag{3.1}$$

The adjusted portfolio $w + x$ is then held for a fixed period of time. At the end of that period, the return on asset i is the random variable a_i. All random variables are on a given probability space, for which $\mathbf{E}$ denotes expectation. We assume knowledge of the first and second moments of the joint distribution of $a = (a_1, \ldots, a_n)$,

$$\mathbf{E}\, a = \bar{a}, \qquad \mathbf{E}(a - \bar{a})(a - \bar{a})^T = \Sigma.$$

A riskless asset can be included, in which case the corresponding $\bar{a}_i$ is equal to its (certain) return, and the ith row and column of Σ are zero.

The end of period wealth is a random variable, $W = a^T(w + x)$, with expected value and variance given by

$$\mathbf{E}\, W = \bar{a}^T(w + x), \qquad \mathbf{E}(W - \mathbf{E}\, W)^2 \;=\; (w + x)^T \Sigma\, (w + x). \tag{3.2}$$

The budget constraint (3.1) can also be written as an inequality,

$$\mathbf{1}^T x + \phi(x) \leq 0. \tag{3.3}$$

With appropriate assumptions ($\bar{a}_i > 0$, $\phi \geq 0$), solving an expected wealth maximization problem with either form of the budget constraint yields the same result. The

inequality form is more appropriate for use with numerical optimization methods. (For example, if ϕ is convex the inequality constraint (3.3) defines a convex set, while the equality constraint (3.1) does not.)

We summarize the portfolio selection problem as

$$\begin{array}{ll} \text{maximize} & \bar{a}^T(w+x) \\ \text{subject to} & \mathbf{1}^T x + \phi(x) \leq 0 \\ & w + x \in \mathcal{S} \end{array} \tag{3.4}$$

where

$\bar{a} \in \mathbf{R}^n$	is the vector of expected returns on each asset,
$w \in \mathbf{R}^n$	is the vector of current holdings in each asset,
$x \in \mathbf{R}^n$	is the vector of amounts transacted in each asset,
$\phi : \mathbf{R}^n \to \mathbf{R}$	is the transaction costs function,
$\mathcal{S} \subseteq \mathbf{R}^n$	is the set of feasible portfolios.

In the next two sections we describe a variety of transaction costs functions ϕ and portfolio constraint sets $\mathcal{S}$.

3.3 Transaction costs

Transaction costs can be used to model a number of costs, such as brokerage fees, bid-ask spreads, taxes, fund loads, or even monitoring costs. In this chapter, we assume the transaction costs to be separable, *i.e.*, the sum of the transaction costs associated with each trade:

$$\phi(x) = \sum_{i=1}^{n} \phi_i(x_i),$$

where ϕ_i is the transaction cost function for asset i.

The simplest model for transaction costs is that there are none, *i.e.*, $\phi(x) = 0$. In this case the original portfolio is irrelevant, except for its total value. We can make whatever transactions are necessary to arrive at the optimal portfolio.

A better model of real transactions costs is a linear one, with the costs for each transaction proportional to the amount traded:

$$\phi_i(x_i) = \begin{cases} \alpha_i^+ x_i, & x_i \geq 0 \\ -\alpha_i^- x_i, & x_i \leq 0. \end{cases} \tag{3.5}$$

Here α_i^+ and α_i^- are the cost rates associated with buying and selling asset i. Linear transaction costs can be used, for example, to model the gap between bid and ask prices.

Since the linear transaction costs functions ϕ_i are convex, the budget constraint can be handled by convex optimization. Specifically, linear costs can be handled by introducing new variables $x^+, x^- \in \mathbf{R}^n$, expressing the total transaction as

$$x_i = x_i^+ - x_i^-,$$

with the constraints $x_i^+ \geq 0$, $x_i^- \geq 0$. The transaction costs function ϕ_i is then represented as

$$\phi_i = \alpha_i^+ x_i^+ + \alpha_i^- x_i^-.$$

Any piecewise linear convex transaction costs function can be handled in a similar way.

In practice, transaction costs are not convex functions of the amount traded. Indeed, the costs for either buying or selling are almost always *concave*. For example, a fixed charge for any nonzero trade is common, and there are often one or more breakpoints above which the transaction costs per share decrease.

We will consider a simple model that includes fixed plus linear costs, but our method is readily extended to handle more complex transaction costs functions. Let β_i^+ and β_i^- be the fixed costs associated with buying and selling asset i. The fixed-plus-linear transaction costs function is given by

$$\phi_i(x_i) = \begin{cases} 0, & x_i = 0 \\ \beta_i^+ + \alpha_i^+ x_i, & x_i > 0 \\ \beta_i^- - \alpha_i^- x_i, & x_i < 0. \end{cases} \tag{3.6}$$

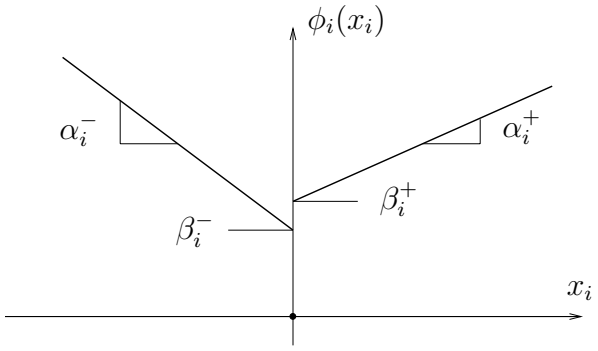

Figure 3.1: Fixed plus linear transaction costs $\phi_i(x_i)$ as a function of transaction amount x_i. There is no cost for no transaction, *i.e.*, $\phi_i(0) = 0$.

which is illustrated in Figure 3.1. Evidently this function is not convex, unless the fixed costs are zero. Therefore, the budget constraint (3.3) cannot be handled by convex optimization.

3.4 Portfolio constraints

3.4.1 Diversity

Constraints on portfolio diversity can be expressed in terms of linear inequalities, and therefore are readily handled by convex optimization.

Individual diversity constraints limit the amount invested in each asset i to a maximum of p_i,

$$w_i + x_i \leq p_i, \qquad i = 1, \ldots, n. \tag{3.7}$$

Alternatively, we can limit the fraction of the total (port transaction) wealth held in each asset:

$$w_i + x_i \leq \gamma_i \mathbf{1}^T (w + x), \qquad i = 1, \ldots, n.$$

These are linear, and therefore convex, inequality constraints on x.

More sophisticated diversity constraints limit the amount of the total wealth that can be concentrated in any small group of assets. Suppose, for example, that we

require that no more than a fraction γ of the total wealth be invested in fewer than r assets. Letting $f_{[i]}$ denote the ith largest component of the vector f, this constraint can be expressed as

$$\sum_{i=1}^{r}(w+x)_{[i]} \leq \gamma \mathbf{1}^T(w+x). \tag{3.8}$$

The left-hand side gives maximum wealth held in any subset of r assets. The right-hand side is a factor γ times the total (post transaction) wealth.

To see that the diversity constraint (3.8) is convex, we can express it as a set of $\begin{pmatrix} n \\ r \end{pmatrix}$ linear inequalities, one for each possible combination of r assets chosen from the n assets. This representation is clearly impractical, however, as this number of linear inequalities can be extremely large.

The diversity constraint (3.8) can be far more efficiently represented by $1 + 2n$ linear inequalities,

$$\begin{aligned} &\gamma \mathbf{1}^T(w+x) \geq rt + \mathbf{1}^T y \\ &t + y_i \geq w_i + x_i, \quad i = 1, \ldots, n, \\ &y_i \geq 0, \qquad\qquad\quad i = 1, \ldots, n, \end{aligned} \tag{3.9}$$

where $y \in \mathbf{R}^n$ and $t \in \mathbf{R}$ are new variables. The proof of equivalence between the two formulations is outlined below.

Several extensions of this type of diversity constraint are possible. For example, we can divide our n assets into N classes of assets, and require that no more than a fraction γ of the total wealth be invested in fewer than R of these classes.

Equivalence of formulations

We outline the proof of equivalence between the diversity-constraint formulations (3.8) and (3.9). The solution of the linear program

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{n} x_i z_i \\ \text{subject to} & 0 \leq z_i \leq 1, \quad i = 1, \ldots, n \\ & \displaystyle\sum_{i=1}^{n} z_i = r, \end{array} \tag{3.10}$$

(where the variable is $z \in \mathbf{R}^n$) is equal to the sum of the r largest components of x, *i.e.*, to $\sum_{i=1}^{r} x_{[i]}$. By strong duality (assuming the problem is strictly feasible and bounded) the optimal value of (3.10) is the same as that of the dual linear program

$$\begin{array}{ll} \text{minimize} & \displaystyle rt + \sum_{i=1}^{n} y_i \\ \text{subject to} & t + y_i \geq x_i, \quad i = 1, \ldots, n \\ & y_i \geq 0, \qquad\quad i = 1, \ldots, n, \end{array} \tag{3.11}$$

(where the variables are $y \in \mathbf{R}^n$ and $t \in \mathbf{R}$). The optimal value of (3.11) is $\sum_{i=1}^{r} x_{[i]}$, and this is less than $\gamma \sum_{i=1}^{n} x_i$ if and only if there exists a feasible solution t, y with $rt + \sum_{i=1}^{n} y_i \leq \gamma \sum_{i=1}^{n} x_i$. This is equivalent to the set of inequalities in (3.9) being feasible.

3.4.2 Shorting

Shorting constraints also lead to linear inequalities. Individual bounds s_i on the maximum amount of shorting allowed on asset i are

$$w_i + x_i \geq -s_i, \qquad i = 1, \ldots, n. \tag{3.12}$$

(In the case of a riskless asset, s_i is a credit line.) If shorting is not permitted, the s_i are set to zero. A bound S on total shorting is

$$\sum_{i=1}^{n} (w_i + x_i)_- \ \leq \ S,$$

where $(\xi)_- = \max\{-\xi, 0\}$. This can be rewritten as a set of linear constraints by introducing an auxiliary variable $t \in \mathbf{R}^n$,

$$\begin{array}{ll} t_i \geq -(w_i + x_i), \qquad t_i \geq 0, \qquad i = 1, \ldots, n \\ \mathbf{1}^T t \leq S. \end{array} \tag{3.13}$$

Other variations can be handled in a similar fashion, among which we mention, for its practical interest, the collateralization requirement

$$\sum_{i=1}^{n} (w_i + x_i)_- \ \leq \ \gamma \sum_{i=1}^{n} (w_i + x_i)_+ ,$$

which limits the total of short positions to a fraction γ of the total of long positions.

3.4.3 Variance

The standard deviation of the end of period wealth W is constrained to be less than $\sigma_{\max}$ by the (convex) quadratic inequality

$$(w + x)^T \Sigma (w + x) \leq \sigma_{\max}^2,$$

which is readily handled by convex optimization. We can also express this constraint as

$$\|\Sigma^{1/2}(w + x)\| \leq \sigma_{\max}, \tag{3.14}$$

where $\|\cdot\|$ is the Euclidean or ℓ_2 norm, and $\Sigma^{1/2}$ is the (symmetric) matrix square root of Σ. The constraint (3.14), which is convex, is a *second-order cone constraint*, and is efficiently handled by recently developed interior-point methods [38].

Equivalently, a maximum $\sigma_{R,\max}$ can be imposed on the standard deviation of

the return R, defined as the ratio of end of period wealth to current wealth, *i.e.*, $R = W/(\mathbf{1}^T w)$. This constraint can be expressed as

$$\|\Sigma^{1/2}(w + x)\| \leq \sigma_{R,\max} \mathbf{1}^T w,$$

which is also a second-order cone constraint.

3.4.4 Shortfall risk

In this section we assume that the returns, the random vector a, have a jointly Gaussian distribution, $a \sim \mathcal{N}(\bar{a}, \Sigma)$. We impose the requirement that the end of period wealth W be larger than some undesired level W^{low} with a probability (or confidence level) exceeding η, where $\eta \geq 0.5$:

$$\mathbf{Prob}\left(W \geq W^{\text{low}}\right) \geq \eta. \tag{3.15}$$

(For $W^{\text{low}} < \mathbf{1}^T w$, this corresponds to a *value at risk* per dollar invested of $\text{VaR} = 1 - W^{\text{low}}/\mathbf{1}^T w$, for a confidence level of η.) We will show that this probability constraint can be expressed as a second-order cone constraint.

The end of period wealth is a Gaussian random variable, $W = a^T(w + x) \sim \mathcal{N}(\mu, \sigma^2)$. The constraint (3.15) can be written as

$$\mathbf{Prob}\left(\frac{W - \mu}{\sigma} \leq \frac{W^{\text{low}} - \mu}{\sigma}\right) \leq 1 - \eta.$$

Since $(W - \mu)/\sigma$ is a zero mean, unit variance Gaussian variable, the probability above is simply $\Phi((W^{\text{low}} - \mu)/\sigma)$, where

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt$$

is the cumulative distribution function of a zero mean, unit variance Gaussian random

variable. Thus, the probability constraint (3.15) can be expressed as

$$\frac{W^{\text{low}} - \mu}{\sigma} \leq \Phi^{-1}(1-\eta),$$

or, using $\Phi^{-1}(1-\eta) = -\Phi^{-1}(\eta)$,

$$\mu - W^{\text{low}} \geq \Phi^{-1}(\eta)\sigma.$$

From $\mu = \bar{a}^T(w+x)$ and $\sigma^2 = (w+x)^T\Sigma(w+x)$, we obtain

$$\Phi^{-1}(\eta)\|\Sigma^{1/2}(w+x)\| \leq \bar{a}^T(w+x) - W^{\text{low}}.$$

Now, provided $\eta \geq 0.5$ (and therefore $\Phi^{-1}(\eta) \geq 0$), this constraint is a second-order cone constraint. (If $\eta < 0.5$, the shortfall risk constraint becomes concave in x.)

It follows that, under the Gaussian assumption, we can impose one or more shortfall risk constraints, and preserve convexity of the problem. We might impose, for example, a constraint on a merely bad return, with some modest confidence, as well as a constraint on a truly disastrous return, with much greater confidence.

Constraints on loss probability have a simple rectangular shape on a figure showing the cumulative distribution of the return, as illustrated in Figure 3.2. In the usual expected return versus standard deviation plane, loss probability constraints define half-plane of allowable points, as illustrated in Figure 3.3.

Optimizing for the expected return under shortfall probability constraints, as described here, is also known as the Telser criterion. We can also solve for the Kataoka criterion, where one shortfall probability constraint is included, and the program objective is changed to maximize W^{low}. This is also a convex problem: the objective is linear and the constraint is a second-order cone. (Another variation is the Roy criterion, where again one shortfall probability constraint is included, but the objective is to maximize the confidence level η. This is not a convex problem. It is, however, a quasiconvex problem, and can be efficiently and globally solved by bisection on η.)

While in the rest of this chapter we only assume knowledge of the first and second moments of the joint distribution of asset returns, this treatment of shortfall risk

constraints requires the assumption of a jointly Gaussian distribution. In practice, the observed returns are seldom Gaussian. They often are skewed, or have "fat tails", *i.e.*, resemble a Gaussian distribution in the central area but have higher probability mass for high deviations. Nevertheless, a shortfall probability approach can be effective if used in an informed manner.

Alternatively, we can again assume no knowledge of the distribution except for the first and second moments, and use the Chebyshev bound to limit the shortfall probability (as already suggested by Roy [56]). In this case, the factor $\Phi^{-1}(\eta)$ in the formulas above is replaced by $(1-\eta)^{-1/2}$.

References for downside-risk approaches include Roy [56], Telser [64], Rudolf [57], Leibowitz *et al.* [34], and Lucas and Klaasen [39].

3.5 Convex portfolio optimization problems

If any number of convex transaction costs and convex constraints are combined, the resulting problem is convex. Linear transaction costs, as well as all the portfolio constraints described above, are convex, indeed, second-order cone programs. Such problems can be globally solved with great efficiency, even for problems with a large number of assets and constraints.

3.5.1 Example

We consider the specific problem

$$
\begin{array}{ll}
\text{maximize} & \bar{a}^T(w + x^+ - x^-) \\
\text{subject to} & \\
& \mathbf{1}^T(x^+ - x^-) + \sum_{i=1}^n (\alpha_i^+ x_i^+ + \alpha_i^- x_i^-) \leq 0 \\
& x_i^+ \geq 0, \quad x_i^- \geq 0, \quad i = 1, \ldots, n \\
& w_i + x_i^+ - x_i^- \geq s_i, \quad i = 1, \ldots, n \\
& \Phi^{-1}(\eta_j) \|\Sigma^{1/2}(w + x^+ - x^-)\| \leq \bar{a}^T(w + x^+ - x^-) - W_j^{\text{low}}, \quad j = 1, \ldots, 2,
\end{array}
$$

with 100 risky and one riskless assets (so $n = 101$). This specifies linear transaction costs, a limit on shorting of s_i per asset, and two shortfall-risk constraints.

The mean and covariance of the risky assets was estimated from one year of daily closing prices of S&P 500 stocks (the first 100, alphabetically by ticker, with a full year of data from January 9, 1998 to January 8, 1999). The distribution was scaled for a portfolio holding period of 20 days. The riskless asset was assumed to have unit return. Far more sophisticated methods could have been used to estimate the return mean and covariance; our only goal here is to demonstrate the optimization method.

For transactions cost and constraint parameters, we (arbitrarily) selected the values

$$\begin{array}{ll} w_1, \ldots, w_{100} = 1/101, & w_{101} = 1/101 \\ \alpha_1^+, \ldots, \alpha_{100}^+ = 1\%, & \alpha_{101}^+ = 0 \\ \alpha_1^-, \ldots, \alpha_{100}^- = 1\%, & \alpha_{101}^- = 0 \\ s_1, \ldots, s_{100} = 0.005, & s_{101} = 0.5, \end{array}$$

(where index 101 is the riskless asset). For the shortfall constraints, we chose

$$\eta_1 = 80\%, \qquad W_1^{\text{low}} = 0.9; \qquad\qquad \eta_2 = 97\%, \quad W_2^{\text{low}} = 0.7,$$

which correspond to a limit on probabilities of a bad and of a disastrous return, respectively.

This problem is a second-order cone program (SOCP), with 202 variables and 306 constraints. The optimal portfolio was obtained in approximately three minutes on a personal computer, using the general purpose software `socp`, which does not take any advantage of the sparsity in the problem data.

Figure 3.2 plots the cumulative distribution of the return for the optimal portfolio. The 50% probability level corresponds, on the horizontal axis, to the expected return. The loss probability constraints are also drawn in the figure. Note that the 0.7 return (0.3 value at risk) for a 97% confidence level is the active constraint.

Figure 3.3 plots the tradeoff curve of expected return versus standard deviation of return, which is the efficient frontier for the problem (ignoring the shortfall probability

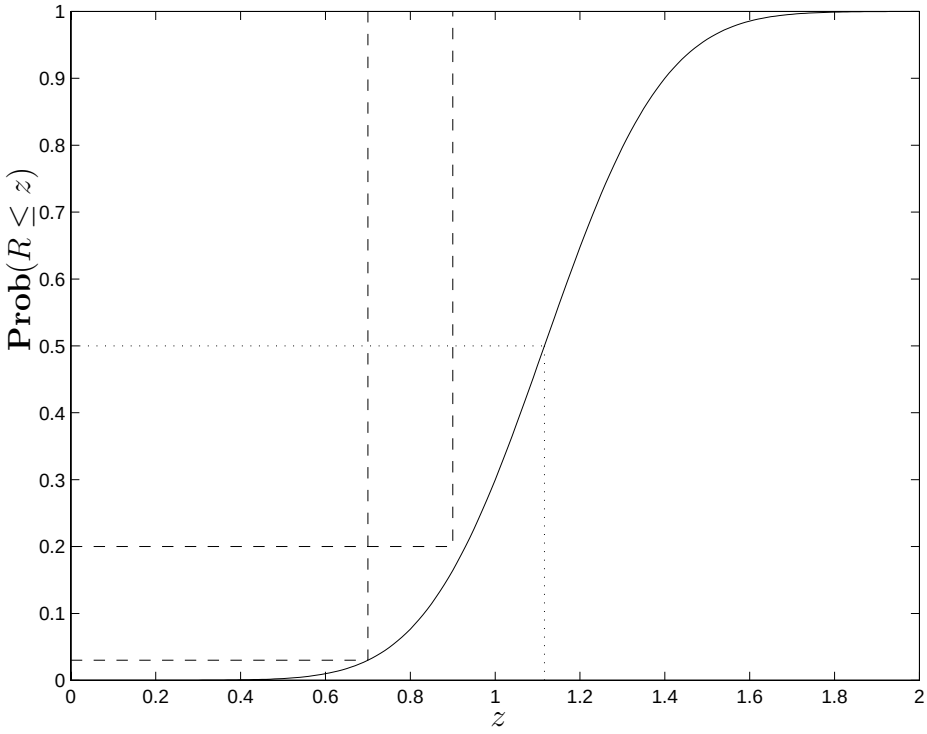

Figure 3.2: Cumulative distribution function of the return, for the optimal portfolio in the example with 100 stocks plus riskless asset. The expected return, which is also the median return since the distribution is assumed Gaussian, is shown with the dotted line. The two limits on probability of shortfall are shown as dashed lines. The limit on the right, and higher, limits the probability of a return below 0.9 (*i.e.*, a bad return) to no more than 20%; the limit on the left, and lower, limits the probability of a return below 0.7 (*i.e.*, a disastrous return) to no more than 3%.

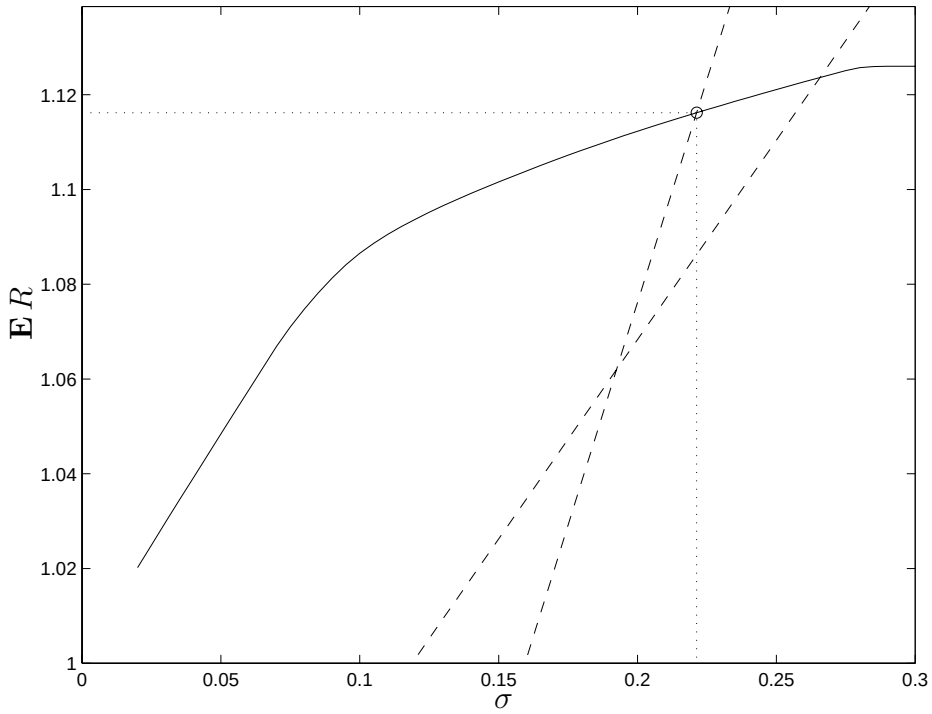

Figure 3.3: Efficient frontier of return mean versus return variance for example problem with 100 stocks plus riskless asset, ignoring the shortfall probability constraints. The sloped dashed lines show the limits imposed by the shortfall probability constraints. The optimal solution of the problem with the shortfall constraints is shown as the small circle.

constraints). On this plot, the shortfall probability constraints correspond to half-planes, whose boundaries are shown as dashed lines in the figure. The dashed line with the smaller slope corresponds to the limit on the probability of the 0.9 return; the line with steeper slope corresponds to the limit on the probability of the disastrous loss. The optimal solution is marked with a small circle.

3.6 Fixed transaction costs

To simplify notation, we assume from now on equal costs for buying and selling, the extension for nonsymmetric costs being straightforward. The transaction costs function is then

$$\phi(x) = \sum_{i=1}^{n} \phi_i(x_i),$$

with

$$\phi_i(x_i) = \begin{cases} 0, & x_i = 0 \\ \beta_i + \alpha_i |x_i|, & x_i \neq 0. \end{cases} \tag{3.16}$$

In the general case, costs of this form lead to a hard combinatorial problem.

The simplest way to obtain an approximate solution is to ignore the fixed costs, and solve for $\phi_i(x_i) = \alpha_i |x_i|$. If the β_i are very small, this may lead to an acceptable approximation. In general, however, it will generate inefficient solutions with too many transactions. (Note that if this approach is taken and the solution is computed disregarding the fixed costs, some margin must be added to the budget constraint to allow for the payment of the fixed costs.)

On the other hand, by considering the fixed costs, we discourage trading small amounts of a large number of assets. Thus, we obtain a *sparse* vector of trades; *i.e.*, one that has many zero entries. This means most of the trading will be concentrated in a few assets, which is a desirable property.

3.6.1 Finding the global optimum

To compute the exact solution, an exhaustive combinatorial search can be performed. Each x_i can either be set to zero (with $\phi_i(x_i) = 0$), or assumed to be nonzero (with $\phi_i(x_i) = \alpha_i |x_i| + \beta_i$). These two alternatives for each of the n assets yields 2^n combinations. For each of these a convex optimization problem results, since the ϕ_i are convex (either zero or linear plus a constant). If we solve each of these 2^n convex problems, the global optimum is found by choosing, out of the 2^n solutions, the one with the highest expected end of period wealth. Because of computational requirements, this approach becomes difficult for n larger than 10, and certainly unrealistic

for n larger than 20.

The exhaustive search can be improved by using branch-and-bound methods, which can greatly reduce the required computational effort. These methods require a procedure for computing upper and lower bounds. We will next propose a heuristic, which can be used to find approximate solutions (and therefore lower bounds). Our experience indicates that this heuristic performs consistently well, producing high-quality suboptimal solutions. We also show how to compute a global upper bound on the achievable end of period wealth, which makes it possible to embed the heuristic in branch-and-bound methods.

3.6.2 Convex relaxation and global bound

We assume that lower and upper bounds on the x_i are known, *i.e.*, l_i and u_i for which x_i must satisfy

$$-l_i \leq x_i \leq u_i.$$

(We will later describe how to obtain such bounds from the portfolio constraints.) The *convex envelope* of ϕ_i, which is the largest convex function which is lower or equal to ϕ_i in the interval $[-l_i, u_i]$, is given by

$$\phi_i^{\text{c.e.}}(x_i) = \begin{cases} \left(\dfrac{\beta_i}{u_i} + \alpha_i\right) x_i\,, & x_i \geq 0 \\ -\left(\dfrac{\beta_i}{l_i} + \alpha_i\right) x_i\,, & x_i \leq 0. \end{cases} \tag{3.17}$$

This is shown in Figure 3.4. The extension for the case when one or both bounds are not available, *i.e.*, $l_i = -\infty$ or $u_i = +\infty$, is trivial (*e.g.*, $\phi_i^{\text{c.e.}}(x_i) = \alpha_i x_i$ for $x_i \geq 0$ if $u_i = +\infty$.)

Using $\phi_i^{\text{c.e.}}$ for ϕ_i relaxes the budget constraint, in the sense that it enlarges the

search set. Consider the portfolio selection problem (3.4), with $\phi_i^{\text{c.e.}}$ replaced for ϕ_i,

$$\begin{array}{ll} \text{maximize} & \bar{a}^T(w+x) \\ \text{subject to} & \mathbf{1}^T x + \sum_{i=1}^n \phi_i^{\text{c.e.}}(x_i) \leq 0 \\ & w + x \in \mathcal{S}. \end{array} \tag{3.18}$$

This corresponds to optimizing the same objective as before (the expected end-of-period wealth), subject to the same portfolio constraints, but with a looser budget constraint. Therefore the optimal value of (3.18) is an upper upper bound on the optimal value of the unmodified problem (3.4). Since the problem (3.18) is convex, we can compute its optimal solution, and hence the upper bound on the optimal value of the original problem (3.4), very efficiently.

Note that in most cases the optimal transactions vector for the relaxed problem (3.18) will not be feasible for the original problem (3.4). The unmodified budget constraint will not be satisfied by the solution of the modified program, except in the very special case when each transaction amount x_i is either l_i, u_i, or 0. (These are the three values for which the convex envelope and the true transaction costs function agree.)

3.6.3 Bounds on the x_i

Upper and lower bounds on the x_i are required to perform the convex relaxation and obtain a global upper bound on the expected end of period wealth. Such bounds are easily derived from the diversity, shorting and budget constraints. However, it is highly desirable to find tighter bounds on the x_i (because the tightness of the global upper bound will in turn depend on the tightness of these bounds). The computation of such bounds depends on the particular problem being addressed. If the problem includes a constraint on the variance of portfolio return, an upper bound on x_i is given by the solution of

$$\begin{array}{ll} \text{maximize} & f(x) = x_i \\ \text{subject to} & g(x) = (w+x)^T \Sigma (w+x) - \sigma^2 \leq 0. \end{array}$$

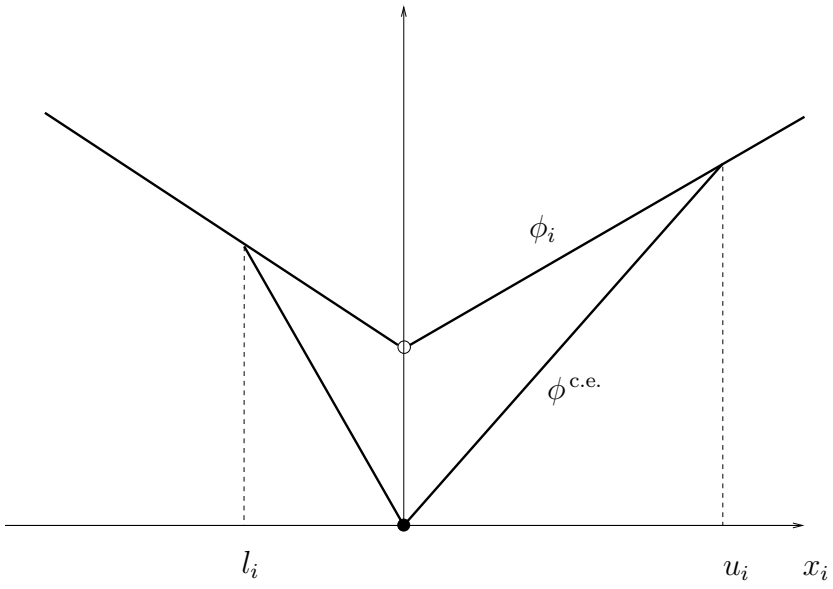

Figure 3.4: The convex envelope of ϕ_i over the interval $[l_i, u_i]$, is the largest convex function smaller than ϕ_i over the interval. For fixed plus linear costs, as shown here, the convex envelope is a linear transaction costs function.

The gradients are

$$\nabla_x f = \lambda e_i, \quad \nabla_x g = 2\Sigma(w + x).$$

Forming the Lagrangian and minimizing, a simple calculation leads to

$$x_i = \sigma\sqrt{(\Sigma^{-1})_{ii}} - w_i,$$

where $(\,\cdot\,)_{ii}$ denotes the i, i element of the matrix. The same can be done for a lower bound,

$$x_i = -\sigma\sqrt{(\Sigma^{-1})_{ii}} - w_i,$$

although, in this case, the bounds derived from the shorting constraints are often observed to be tighter.

3.6.4 Iterative heuristic

We now describe a heuristic for finding a feasible, suboptimal portfolio, which is based on the same method used to find an upper bound. At the end of that section, we noted that the transaction x that is optimal for the relaxed problem (3.18) would be feasible for the original problem (3.4) if, by chance, the optimal x for the relaxation satisfied $\phi^{\text{c.e.}}(x) = \phi(x)$. This only happens when each transaction is either zero, or at one of its bounds, l_i or u_i.

The iterative procedure uses a modified transaction costs function which, like the relaxed cost function, is convex. Unlike the relaxed cost function, however, we do not require the convex cost function to be a lower bound on the true transaction costs function.

An iterated reweighting of this convex cost function is used, in such a way that most small transactions in the solution are driven to zero. We use δ (some nonnegative and small value) as a threshold for deciding when a transaction is considered to be zero. Since each of these reweighted modified functions is convex, each iteration consists in solving a convex program. The feasibility of the portfolio is obtained by ensuring that the modified transaction costs function $\widehat{\phi}_i^k$ agrees with the true ϕ_i at the solution transactions x_i^*. These ideas will become clearer with the ensuing discussion.

Consider the following procedure.

1. $k := 0$.
   Solve the convex relaxed problem (3.18).
   Let x^0 be the solution to this problem.

2. $k := k + 1$.
   Given the solution to the previous problem x^{k-1}, define $\widehat{\phi}_i^k$ as

$$\widehat{\phi}_i^k(x_i) = \left(\frac{\beta_i}{|x_i^{k-1}| + \delta} + \alpha_i\right)|x_i|.$$

   Solve the modified (convex) portfolio selection problem

$$\begin{array}{ll} \text{maximize} & \bar{a}^T(w + x) \\ \text{subject to} & \mathbf{1}^T x + \sum_{i=1}^n \widehat{\phi}_i^k(x_i) \leq 0 \\ & w + x \in \mathcal{S}. \end{array} \tag{3.19}$$

   Let x^k be the solution to this problem.

3. If the portfolios x^k and x^{k-1} found in the two previous iterations are (approximately) equal, return $x^* := x^k$ and exit.
   Otherwise, go to step 2.

A rough interpretation of the algorithm is that in each iteration we amortize the fixed costs evenly over the transaction amount in the previous iteration.

If this iterative procedure exits, which occurs if two successive iterates are close to each other, the solution x^* will be nearly feasible for the original problem (see Figure 3.5). This is seen by noting that, for $x_i^* \gg \delta$,

$$\widehat{\phi}_i(x_i^*) \;=\; \left(\frac{\beta_i}{|x_i^*| + \delta} + \alpha_i\right)|x_i^*| \;\approx\; \beta_i + \alpha_i|x_i^*| \;=\; \phi_i(x_i^*),$$

and for $x_i^* = 0$,

$$\widehat{\phi}_i(x_i^*) \;=\; 0 \;=\; \phi_i(x_i^*).$$

In a sense, δ defines a *soft threshold* for deciding whether a given x_i^* is considered zero, *i.e.*, whether the corresponding transaction should be performed or not. In a practical implementation of the portfolio trades, a hard threshold is needed, and the x_i^* on the order of δ or smaller should be taken as zero. Note that while $\widehat{\phi}_i(x_i^*) \leq \phi_i(x_i^*)$ for all x_i^*, this inequality is tight except for x_i^* in the order of δ. Such terms may lead to feasibility problems, with a nonnegligible violation of the budget constraint. In practice this is not an issue since terms in the order of δ will seldom appear in the solution. On each iteration, for the x_i^k that become small, the modified $\widehat{\phi}_i^{k+1}(x_i)$ has an increased derivative. This eventually pushes the small x_i to zero, leading to sparse solutions.

This also provides the motivation for the method, and an intuition to justify the quality of the approximate solutions that have been found in numerical experiments. (We note in Appendix 3.6.5 that this heuristic is equivalent to finding a local minimum of a log-like, concave function.) A proof of convergence for the heuristic is given in Appendix 3.6.5. Our numerical experiments indicate that convergence occurs in about 4 iterations or less for a wide range of problems.

Upper and lower bounds on the global optimum for the expected end of period wealth are given by $\bar{a}^T x^0$ and $\bar{a}^T x^*$. As a final step, an extra iteration can be included with the sparsity pattern fixed, and with the transaction costs exact for that pattern. That is, the small x_i are set to zero, with $\phi_i(x_i) = 0$, and the others are assumed nonzero, with $\phi_i(x_i) = \beta_i + \alpha_i|x_i|$. (This is equivalent to one particular combination, out of the possible 2^n in the exhaustive search.)

3.6.5 Convergence of the heuristic

The heuristic described performs successive relaxations of a nonconvex constraint. For simplicity, we outline the proof for a similar successive relaxation of a nonconvex objective function (the results are easily extended). Also for simplicity, assume $x_i \geq 0$. Define the transformation $A : \mathbf{R}^n \to \mathbf{R}^n$,

$$A(y) = \operatorname*{arginf}_{x \in \mathcal{S}} \sum_{i=1}^{n} \frac{x_i}{y_i + \delta},$$

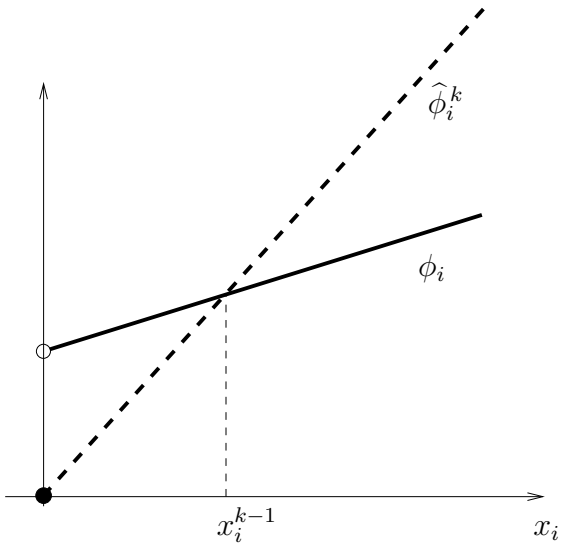

Figure 3.5: One iteration of the algorithm. Each of the nonconvex transaction costs (plotted as a solid line) is replaced by a convex one (plotted as a dashed line) that agrees with the nonconvex one at the current iterate. If two successive iterates are the same, then the iterates are feasible for the original nonconvex problem.

with $\mathcal{S} \subset \mathbf{R}^n$ a convex, compact set, and $\delta > 0$. We show that the sequence x^k such that

$$x^0 \in \mathcal{S}, \qquad x^{k+1} = A(x^k),$$

satisfies $x_i^{k+1} - x_i^k \to 0$, for $i = 1, \ldots, n$.

To prove this result, we first define the function $L : \mathbf{R}^n \to \mathbf{R}_+$,

$$L(x) = \prod_{i=1}^{n} (x_i + \delta),$$

and show that the sequence $L(x^k)$ is monotonically nonincreasing (accordingly, we refer to L as the *descent function.*) Since $x^{k+1} = A(x^k)$ yields the infimum over $\mathcal{S}$, and $x^k \in \mathcal{S}$, we have that

$$\sum_{i=1}^{n} \frac{x_i^{k+1} + \delta}{x_i^k + \delta} \;\leq\; \sum_{i=1}^{n} \frac{x_i^k + \delta}{x_i^k + \delta} \;=\; n.$$

Using the inequality between the arithmetic and geometric means for nonnegative terms, we conclude that

$$\prod_{i=1}^{n} \frac{x_i^{k+1} + \delta}{x_i^k + \delta} \leq 1,$$

which implies $L(x^{k+1}) \leq L(x^k)$. Since L is bounded below by δ^n, the sequence $L(x^k)$ converges.

Now, convergence of $L(x^k)$ to a nonzero limit implies that

$$\frac{L(x^{k+1})}{L(x^k)} \;=\; \prod_{i=1}^{n} \frac{x_i^{k+1} + \delta}{x_i^k + \delta} \;\to\; 1.$$

Define y^{k+1} to be

$$y_i^{k+1} = \frac{x_i^{k+1} + \delta}{x_i^k + \delta},$$

and write $y_1^k = 1 + \epsilon$. It follows that

$$\prod_{i=1}^{n} y_i^k \;=\; (1+\epsilon) \prod_{i=2}^{n} y_i^k \;\leq\; (1+\epsilon)\left(1 - \frac{\epsilon}{n-1}\right)^{n-1} \;=\; f(\epsilon),$$

where we used $\sum_{i=1}^{n} y_i^{k+1} \leq n$, and the inequality between the arithmetic and geometric means. The function f is continuous in ϵ and, with some algebra, it is easily checked that $f(0) = 1$, $f'(0) = 0$, and $f''(\epsilon) < 0$, for $|\epsilon| < 1$. Therefore, $f(\epsilon) < 1$ for $\epsilon \neq 0$, $|\epsilon| < 1$.

We conclude that $\prod_{i=1}^{n} y_i^k \to 1$ implies $f(\epsilon) \to 1$, and that this in turn implies $\epsilon \to 0$. Hence $y_1^k \to 1$, and likewise for all y_i^k.

Using $x_i^k + \delta \leq M < \infty$ for all k (since the set $\mathcal{S}$ is bounded), we obtain the desired result:

$$x_i^k - x_i^{k+1} \to 0.$$

Note that, upon convergence, the partial derivative with respect to x_i of the function minimized in the last iteration is given by

$$\frac{1}{x_i^* + \delta},$$

which is equal to the derivative of the function

$$\sum_{i=1}^{n} \log(x_i + \delta),$$

at $x_i = x_i^*$. From the equality of the first-order conditions for optimality, it is easy to see that the iterative procedure finds a local minimum in $\mathcal{S}$ of this logarithmic function.

3.7 Examples with fixed costs

For numerical examples, we use the same stock data as in the previous example. We specify fixed plus linear transaction costs, and constraints on shorting and on variance. We first describe an example with 10 stocks, plus a riskless asset. The

parameters used (again, arbitrarily) are

$$\begin{aligned} &w_1, \ldots, w_{11} = 1/11 \\ &\alpha_1, \ldots, \alpha_{10} = 1\%, \qquad \alpha_{11} = 0 \\ &\beta_1, \ldots, \beta_{10} = 0.01, \qquad \beta_{11} = 0 \\ &s_1, \ldots, s_{10} = 0.05, \qquad s_{11} = 0.5. \end{aligned}$$

The small size of this problem allows us to compute the exact solution, that is the global optimum, by combinatorial search. Figure 3.6 displays the resulting trade-off curve, with expected return plotted against standard deviation of return. Four curves are shown: the upper bound; the exact solution; the heuristic solution; and the solution computed without regard for fixed cost. Note that the upper bound is close to the heuristic solution. Note also that the heuristic nearly coincides with the exact solution, even though the heuristic required only about one-thousandth the computational effort. For the heuristic, we used $\delta = 10^{-3}$ (and did not include a final iteration with fixed sparsity pattern).

In Figure 3.7, still for the same 11 assets example, $\sigma_{\max}$ was kept constant at 0.15, and the problem was solved for a range of fixed costs β. The optimal expected return is plotted as a function of fixed costs, with the four curves obtained by the same procedure as in the previous figure. Again we can see that the difference between our heuristic and the optimal is very small. In this figure we can also see the cost of ignoring the transaction costs, which, naturally, increases with increasing fixed transaction costs.

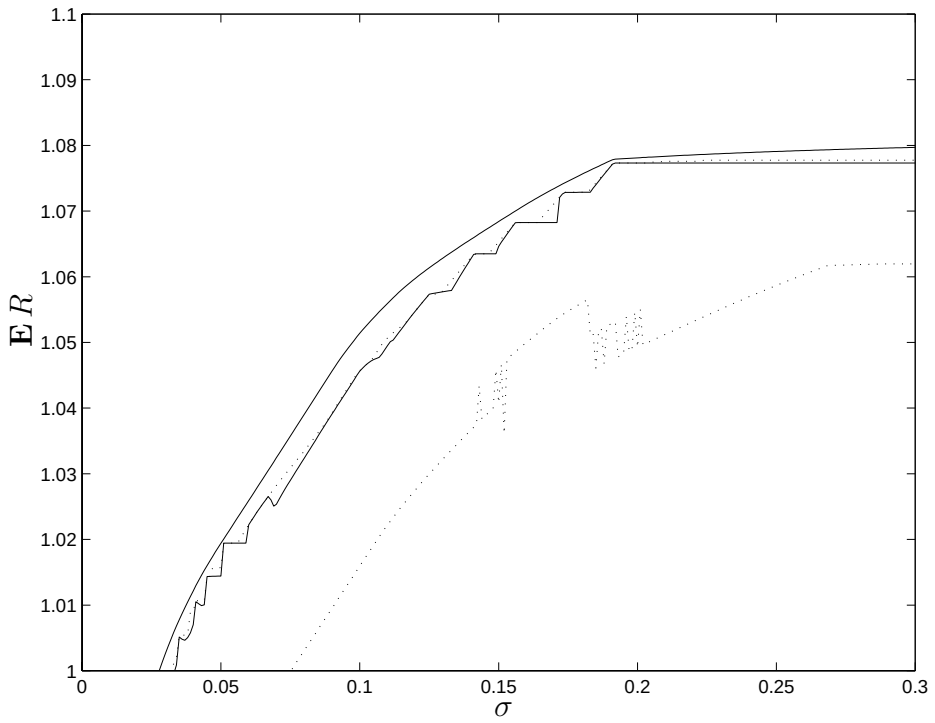

Figure 3.6: Example with 10 stocks plus riskless asset, plot of expected return as a function of standard deviation. Curves from top to bottom are: 1. global upper bound (*solid*), 2. true optimum by exhaustive search (*dotted*), 3. heuristic solution (*solid*), and 4. solution computed without regard for fixed costs (*dotted*). Note that curves 2 and 3 are nearly coincidental.

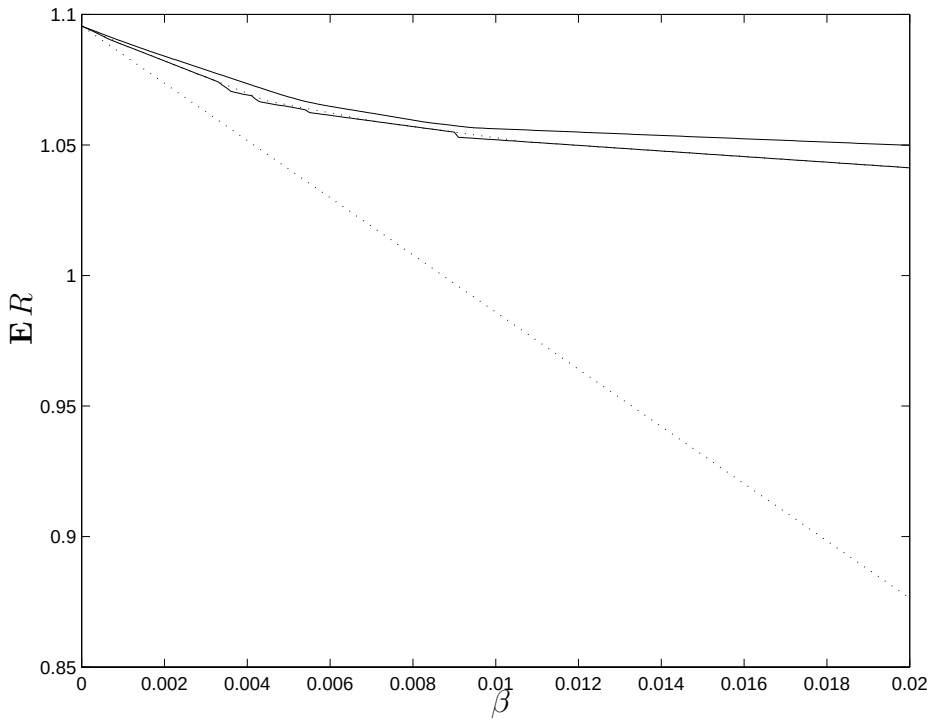

Figure 3.7: Example with 10 stocks plus riskless asset, plot of expected return as a function of fixed transaction costs. Curves from top to bottom are: 1. global upper bound (*solid*), 2. true optimum by exhaustive search (*dotted*), 3. heuristic solution (*solid*), and 4. solution computed without regard for fixed costs (*dotted*). Note that curves 2 and 3 are nearly coincidental.

As a second and larger example, we considered 100 stocks, plus a riskless asset, using the parameters

$$\begin{aligned} &w_1, \ldots, w_{101} = 1/101, \\ &\alpha_1, \ldots, \alpha_{100} = 1\%, \qquad && \alpha_{101} = 0 \\ &\beta_1, \ldots, \beta_{100} = 0.001, \qquad && \beta_{101} = 0 \\ &s_1, \ldots, s_{100} = 0.005, \qquad && s_{101} = 0.5. \end{aligned}$$

Figure 3.8 displays the resulting tradeoff curve. The curves shown are the upper bound, the heuristic solution, and the solution computed without regard for fixed costs. The exact tradeoff curve is not shown since, in this case, it would require a prohibitive effort to compute. However, the fact that the upper bound and the heuristic are close to each other establishes that computing the exact, globally optimal solution would only yield a small improvement over the heuristic solution. Note that the heuristic takes only a few minutes to compute, while the time required to perform the combinatorial search with the same computational resources would be of the order of 2^{100}, or 10^{30} minutes. This is a rather long time for what is guaranteed to be only a marginal improvement. (For reference, the age of the universe is approximately 10^{16} minutes.)

Again, we note that the upper bound and the heuristic can be embedded in a branch-and-bound algorithm to find the global optimum. Since the upper bound and the heuristic are often close, it can be expected that such a branch-and-bound search would exhibit good convergence properties.

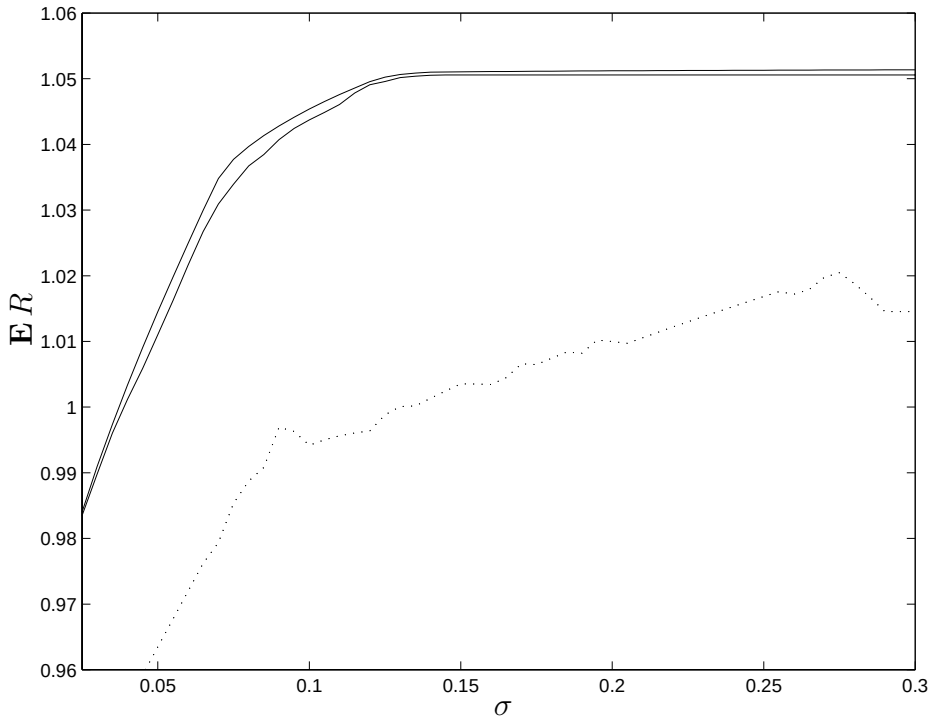

Figure 3.8: Example with 100 stocks plus riskless asset, plot of expected return as a function of standard deviation. Curves from top to bottom are: 1. global upper bound (*solid*), 2. heuristic solution (*solid*), and 3. solution computed without regard for fixed costs (*dotted*).

3.8 Related problems

A related problem is that of tracking a portfolio over a single period, with high accuracy and low cost. This arises, for instance, in the problem of reproducing an index formed by a large number of stocks, where trading in all the relevant stocks would incur excessive costs. In this case, the desired solution is a portfolio consisting of a relatively small subset of the stocks present in the index that, with high probability, behaves like the index. (An equivalent problem is that of portfolio insurance, where the goal is to reduce the downside risk as much as possible by buying put options on a small number of the assets.)

A measure of closeness between the adjusted portfolio $w + x$ and a reference portfolio v is given by the expected square difference in returns (v may be the index to be tracked, or the portfolio to be insured). The expected square difference in the returns of the two portfolios is equal to

$$\mathbf{E}\,(a^T(w+x) - a^T v)^2 = (w + x - v)^T(\Sigma + \bar{a}\bar{a}^T)(w + x - v).$$

We'll refer to this measure as the tracking error, for short. It has a number of interesting properties, but the one of main concern here is that it is convex in x.

The problem can be seen as having two conflicting objectives. In addition to low transaction costs $\phi(x)$, we want a portfolio $w + x$ with small tracking error. We can address this problem in any of the alternative formulations: minimize the costs subject to a bound on the tracking error; minimize the tracking error subject to a bound on the costs; or minimize a weighted combination of costs and tracking error. (Other constraints, such as on budget and shorting, will also be included, of course.)

For linear transaction costs, each of these problem formulations is convex and easy to solve (either a quadratic program or a quadratically constrained quadratic program).

When fixed transaction costs are present, a difficult combinatorial program results. This can be addressed by the same method as in §3.6, which will produce a global bound on achievable performance, and a suboptimal solution. Numerical simulations for the tracking problem have produced suboptimal solutions of consistently

high quality. Nevertheless, if so desired, the global bound and the heuristic can be embedded in a branch-and-bound method to guarantee a high accuracy solution.

Chapter 4

The worst-case risk of a portfolio

4.1 Introduction

Consider a portfolio of risky assets held over a single period, where the distribution of the asset returns is imprecisely known. That is, estimates exist for the expected value and covariance matrix of the asset returns, but these estimates are subject to estimation errors, and possibly to modeling errors.

Modeling errors arise, for example, because statistical procedures make the unrealistic assumption of distributions being stationary. Since the covariance matrix has $n\left(n+1\right)/2$ independent entries, if a large number of assets is to be considered then a very large number of samples is required in order to have good, uncorrelated estimates of all the entries in the covariance matrix. Also, given the usual volatility levels, the estimate of the mean converges slowly and, with any realistic amount of non-stationarity, a good estimate can never be reached. Other pricing models, such as CAPM [62, 35], are also subject to modeling errors.

We use the following notation throughout the chapter. There are n assets, with expected return $\mu \in \mathbf{R}^n$ and covariance $\Sigma \in \mathbf{R}^{n\times n}$. The portfolio under analysis is described by the weight vector $w \in \mathbf{R}^n$, where w_i represents the fraction of the total wealth held in asset i.

In a simple analysis of the risk associated with the portfolio, these estimates of asset statistics are assumed exact. The portfolio expected return and variance are

then assumed to be $\mu^T w$ and $w^T \Sigma w$. Such an approach does not account for the imprecision in the estimates of the asset statistics, which may have a significant effect on the risk associated with the portfolio.

Repeating the analysis under a small number of different scenarios (*i.e.*, with different values for the expected returns and covariance matrix) is a simple way of dealing with inaccuracy in the estimates. It is, however, an inadequate approach for problems with a moderate to large number of assets. A large number of scenarios may be run in a Monte Carlo procedure, but, as the number of assets becomes large, obtaining accurate results by this method quickly becomes numerically too expensive. (Further, while Monte Carlo analysis can work well in analyzing the risk of moderately large portfolios, it is not easily incorporated in an optimization procedure with the purpose of finding a portfolio with desirable characteristics.)

The purpose of this chapter is to present a new approach for upper bounding the risk associated with a portfolio, for a given description of the uncertainty in the estimates of the first and second moments of the asset returns. We also show how to design portfolios that are robustly optimal, in the sense that they minimize this upper bound on risk. In fact, solving portfolio optimization problems with great precision, when the problem parameters (say μ and Σ) are not precisely known is not a reasonable proposition. A better approach is to explicitly account for such parameter uncertainty in the optimization, and to design a portfolio that performs reasonably for any set of parameters within the range of parameter uncertainty.

Consider an example of maximum risk analysis. Given an entry-wise description of uncertainty in the estimate of Σ, how large can the portfolio variance be? Assume we have the following information. We know an upper and lower bound on the variances of each asset, and an upper and lower bound on the covariances of each pair of assets. That is, for each entry Σ_{ij} of the covariance matrix we have an upper bound $\overline{\Sigma}_{ij}$ and a lower bound $\underline{\Sigma}_{ij}$. And, of course, we know that the covariance matrix must be

positive semidefinite. The problem is then

$$\begin{array}{ll} \text{maximize} & w^T \Sigma\, w \\ \text{subject to} & \underline{\Sigma}_{ij} \leq \Sigma_{ij} \leq \overline{\Sigma}_{ij}, \quad i,j = 1,\ldots,n \\ & \Sigma \succeq 0, \end{array} \tag{4.1}$$

where $w \in \mathbf{R}^n$ is fixed, and $\Sigma \in \mathbf{R}^{n\times n}$ is the problem variable.

This is a convex program, with linear objective, and convex constraints. The positive semidefinite constraint is nonlinear and non-differentiable. The problem is, in fact, a *semi-definite program* (SDP), which we introduced in Chapter 1, for which new optimization methods have been developed in recent years. With these methods, the global solution of many convex programs, including SDPs, can be efficiently computed. The computational effort is shown to grow polynomially with problem size, even for nonlinear, non-differentiable problems, making the methods suitable for large problems.

We are, in effect, departing from a Bayesian approach, in which the uncertainty about the distribution would be incorporated by defining a new distribution on the returns (with a larger variance.) An approach similar to the one we discuss here has, however, been used with great success in the field of robust control.

In §4.2 we describe three worst-case portfolio analysis problems that can be solved via numerically efficient convex optimization methods. In §4.3 and §4.4 we look at other descriptions of the uncertainty set for the mean and covariance matrix that are convex and can be efficiently handled. In §4.5 we look at solution methods for the analysis problem. In additions to the general approach via interior-point methods, specialized projection methods can also be used. In §4.6 we describe the corresponding design problem, that is, finding a portfolio that has good worst-case performance. While software that can be used to solve the analysis problem is now widely available, the design problem requires specialized methods. In §4.7 we describe how the design problem can be efficiently solved using cutting plane methods.

4.1.1 Related work

For references on portfolio optimization in the mean-variance setting and on the downside risk approach, we refer to §3.1.1 [43, 56, 57, 41, 34]. The RiskMetrics technical document [46], by J. P. Morgan and Reuters, provides an introductory overview of portfolio risk analysis.

With the increased use of downside risk approaches, more attention is being paid to the effects of uncertainty in the covariance matrix estimates. A recent article by Ju and Pearson [30] in the *Journal of Risk* is titled "Using value-at-risk to control risk taking: how wrong can you be?" This chapter provides the tools to answer this question, and proposes robust optimization as a methology to deal with it.

4.2 The analysis problem

Assume the following data: a) the vector $w \in \mathbf{R}^n$ specifies the weights of the portfolio under analysis; b) the sets $\mathcal{M} \subset \mathbf{R}^n$ and $\mathcal{S} \subset \mathbf{R}^{n \times n}$ define the range of uncertainty for the estimates of the mean and covariance matrix of asset returns, that is, they define the values that μ and Σ may take. Consider the following problems.

- The *variance problem*: compute the worst-case variance of portfolio w, *i.e.*

$$\sup_{\Sigma \in \mathcal{S}} \quad w^T \Sigma w. \tag{4.2}$$

- The *downside risk problem*: compute the worst-case value at risk in portfolio w, for a confidence level of η, *i.e.*

$$\sup_{\mu \in \mathcal{M}, \Sigma \in \mathcal{S}} \quad \gamma \left(w^T \Sigma w\right)^{1/2} - \mu^T w - 1, \tag{4.3}$$

  with $\eta = \Phi(\gamma)$, where $\Phi(\cdot)$ is the c.d.f. of a zero norm, unit variance Gaussian random variable.

- The *tracking error problem*: compute the worst-case expected square tracking

error of portfolio w relative to a reference portfolio with weights v, *i.e.*

$$\sup_{\mu\in\mathcal{M},\Sigma\in\mathcal{S}} \; (w-v)^T \left(\Sigma + \mu\mu^T\right)(w-v). \tag{4.4}$$

Note that we have assumed the uncertainty description for Σ and μ to be separable. Although we won't discuss it in this thesis, this doesn't have to be the case, and some joint descriptions can also be handled. As a second side note, the interpretation of problem (4.3) as the *value at risk* for a confidence level of η requires the assumption of jointly Gaussian distribution of returns. With proper care and adequate interpretation, the downside risk problem can still be used without this assumption. The derivations for problems 4.3 and 4.4 can be found, *e.g.*, in Lobo, Fazel and Boyd [36].

The sets $\mathcal{M}$ and $\mathcal{S}$ can be obtained from statistic considerations (*i.e.*, from confidence intervals), and from assumptions about modeling errors. The process by which $\mathcal{M}$ and $\mathcal{S}$ are obtained, however, is not discussed in this thesis. Our focus is on identifying descriptions of $\mathcal{S}$ and $\mathcal{M}$ that can be handled with new efficient convex optimization methods. In particular, one critical element in any description of the set $\mathcal{S}$ is that all its elements must be positive semidefinite, since they must be valid covariance matrices. In this chapter we show how to handle problems with such a positivity constraint on Σ, as well as with other types of constraints, including box constraints on Σ and μ and ellipsoidal constraints on μ.

4.3 Uncertainty sets for the expected returns vector

The expected asset returns appear as a linear term in the downside risk problem, and as a quadratic term in the tracking error problem. We now show how to obtain solutions for the worst-case analysis of these terms, for two different types of uncertainty sets. In several cases an analytical solution exists.

4.3.1 Box constraint

Consider a set of entry-wise upper and lower bounds on μ (or box constraint, or ℓ_∞ constraint), *i.e.*

$$\underline{\mu}_i \leq \mu_i \leq \overline{\mu}_i, \quad i = 1, \ldots, n.$$

With this constraint specifying the uncertainty in the expected asset returns, analytical solutions are easily derived for the terms in μ that appear in the worst-case problems. For the downside risk problem, which is linear in μ, the solution is:

$$\sup_{\mu \in \mathcal{M}} -\mu^T w \;=\; \overline{\mu}^T (w)_- - \underline{\mu}^T (w)_+,$$

where $(w)_+$ is a vector with entries $\max\{w_i, 0\}$, and $(w)_-$ is a vector with entries $\max\{-w_i, 0\}$. For the tracking error problem, which is quadratic in μ, the solution is:

$$\begin{aligned} &\sup_{\mu \in \mathcal{M}} (\mu^T (w - v))^2 \;= \\ &\quad = \; \max \left\{ \left(\overline{\mu}^T (w - v)_+ - \underline{\mu}^T (w - v)_- \right)^2, \left(\underline{\mu}^T (w - v)_+ - \overline{\mu}^T (w - v)_- \right)^2 \right\}. \end{aligned}$$

4.3.2 Ellipsoidal constraint

A more interesting case is when $\mathcal{M}$ is an ellipsoid, *i.e.*,

$$\mathcal{M} = \{\mu : (\mu - \bar{\mu})^T S^{-1} (\mu - \bar{\mu}) \leq 1\}.$$

Note that for many statistical procedures M will indeed be an ellipsoid, with S collinear with Σ (*i.e.*, a scaled version). This is not the case if μ is inferred by some other process (such as some pricing model, or from information from analysts' reports). We now show how to incorporate worst-case ellipsoidal uncertainty in μ in a practical optimization program. We do not require an assumption of collinearity with Σ.

For the downside risk problem, which is linear in μ, we can easily compute the

worst-case value analytically.

$$\lambda \;=\; \sup_{\mu\in\mathcal{M}} -\mu^T w \;=\; -\inf_{\mu\in\mathcal{M}} \mu^T w \;=\; -\inf_{\|S^{-1/2}\tilde{\mu}\|\leq 1} (\bar{\mu}+\tilde{\mu})^T w \;=\; -\bar{\mu}^T w - \inf_{\|z\|\leq 1} z^T S^{1/2} w,$$

where $S^{1/2}$ is the (symmetric, positive-semidefinite) matrix square root of S. The z that achieves the infimum is easily seen to be

$$z^* = -\frac{S^{1/2}w}{\|S^{1/2}w\|},$$

so that the value of the supremum is

$$\lambda \;=\; -\bar{\mu}^T w + \frac{w^T S w}{\sqrt{w^T S w}} \;=\; -\bar{\mu}^T w + \sqrt{w^T S w} \;=\; -\bar{\mu}^T w + \|S^{1/2}w\|.$$

Therefore, the solution to the downside risk problem (4.3) with ellipsoidal uncertainty in μ is obtained by adding the constant

$$-\bar{\mu}^T w + \|S^{1/2}w\| - 1$$

to the solution of the worst-case variance problem (4.2).

For the tracking error problem, which is quadratic in μ, we can also obtain a numerically tractable program. The derivation is bit more involved. To obtain the worst-case error we need to evaluate

$$\lambda = \sup_{\mu\in\mathcal{M}} (w-v)^T \mu\mu^T (w-v).$$

It is easily seen that λ is the smallest number that satisfies

$$\mu^T (w-v)(w-v)^T \mu \leq \lambda \quad \text{for all } \mu \text{ such that} \quad (\mu-\bar{\mu})^T S^{-1} (\mu-\bar{\mu}) \leq 1. \qquad (4.5)$$

By using the $\mathcal{S}$-procedure [14, p.23], λ satisfies (4.5) if and only if there is a τ such

that

$$\left[\begin{array}{cc} -(w-v)(w-v)^T & 0 \\ 0 & \lambda \end{array}\right] - \tau \left[\begin{array}{cc} -S^{-1} & S^{-1}\bar{\mu} \\ \bar{\mu}^T S^{-1} & 1-\bar{\mu}^T S^{-1}\bar{\mu} \end{array}\right] \succeq 0, \quad \tau \geq 0.$$

This can be rewritten as

$$\left[\begin{array}{cc} \tau S^{-1} & -\tau S^{-1}\bar{\mu} \\ -\tau\bar{\mu}^T S^{-1} & \tau(\bar{\mu}^T S^{-1}\bar{\mu}-1)+\lambda \end{array}\right] - \left[\begin{array}{c} w-v \\ 0 \end{array}\right] \left[\begin{array}{cc} (w-v)^T & 0 \end{array}\right] \succeq 0, \quad \tau \geq 0,$$

and, with the Schur complement [14, p.7], we get the equivalent inequality

$$\left[\begin{array}{ccc} \tau S^{-1} & -\tau S^{-1}\bar{\mu} & w-v \\ -\tau\bar{\mu}^T S^{-1} & \tau(\bar{\mu}^T S^{-1}\bar{\mu}-1)+\lambda & 0 \\ (w-v)^T & 0 & 1 \end{array}\right] \succeq 0, \quad \tau \geq 0.$$

This last formulation involves a positive semidefinite constraint, or linear matrix inequality, and can be efficiently and globally handled [68].

4.4 Uncertainty sets for the covariance matrix

We now turn to the specification of $\mathcal{S}$, that is, to the description of our knowledge of Σ (or, more accurately, the description of the uncertainty in our knowledge of Σ).

The variance and tracking error problems (4.2) and (4.4) are linear in Σ. The downside risk problem (4.3) can be cast as a program where Σ appears linearly:

$$\begin{array}{ll} \text{minimize} & t - \mu^T w \\ \text{subject to} & t^2 \leq \gamma^2\, w^T \Sigma w \\ & \mu \in \mathcal{M},\ \Sigma \in \mathcal{S}, \end{array}$$

where $t \in \mathbf{R}$ is an extra variable. The extra inequality is convex quadratic in t, and linear in Σ.

For a practical optimization method to be effective we want, if at all possible, a

description of the uncertainty set $\mathcal{S}$ that leads to a convex program. All the problems in consideration are (or can be made) linear in Σ, and they will be convex if the set $\mathcal{S}$ is convex.

We have already introduced a positivity constraint and box constraints, both of which are convex. We represent the constraint that Σ be positive semidefinite by

$$\Sigma \succeq 0.$$

This constraint is required to ensure that all the Σ in the uncertainty set $\mathcal{S}$ are valid covariance matrices. It may, however, be omitted: a) if the other constraints are shown to define a subset of the positive semidefinite cone; or b) if Σ is parameterized in such a way that it is guaranteed to always be positive semidefinite (as will be discussed for factor models in §4.4.4). The most straightforward description of an uncertainty set for Σ is by adding to the positivity constraint a set of box constraints, on each entry Σ_{ij}:

$$\underline{\Sigma}_{ij} \leq \Sigma_{ij} \leq \overline{\Sigma}_{ij}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, i$.

4.4.1 Second-moment of the Wishart distribution

If the returns are normally distributed, the estimate of covariance matrix has a Wishart distribution. The second-moment of the Wishart distribution can be used to define a region of confidence for the estimate. The moments of the Wishart distribution are given, *e.g.*, in Muirhead's book [47, p.90]. This leads to an ellipsoidal constraint on the entries of the covariance matrix, *i.e.*,

$$(s - s_0)^T Q (s - s_0) \leq 1$$

where $s \in \mathbf{R}^{n(n+1)/2}$ is a vector representation of the (upper triangular) entries of Σ. This can also be written as a second-order cone constraint, using a square-root of Q. The size of this constraint can be quite large; Q has about $n^4/4$ entries. Note, however, that the matrix Q has a great deal of structure (again, see [47]). Further

work is required to determine how to exploit this structure for fast algorithms.

The region of confidence can be scaled by a factor that specifies the "conservativeness" (large factor) or "aggressiveness" (small factor) of the analysis. Under a probabilistic interpretation, the size of the ellispsoid corresponds to the confidence level, *i.e.*, to the probability of the covariance matrix being in the uncertainty set.

4.4.2 Constraints on the correlation coefficients

A more natural approach may be to write constraints not on the covariances, but on the correlation coefficents. The constraint for coefficient ij,

$$\underline{\rho}_{ij} \leq \rho_{ij} \leq \overline{\rho}_{ij},$$

is equivalent to

$$\underline{\rho}_{ij}\sqrt{\Sigma_{ii}\Sigma_{jj}} \leq \Sigma_{ij} \leq \overline{\rho}_{ij}\sqrt{\Sigma_{ii}\Sigma_{jj}}.$$

If Σ_{ii} and Σ_{jj} are constant (*i.e.*, if the variance of each asset is known exactly), this constraint is linear in Σ. In this case, it is equivalent to a box constraint.

In the general case, however, the constraint on the correlation coefficient is convex only if $\underline{\rho}_{ij} \leq 0$ and $\overline{\rho}_{ij} \geq 0$. In particular, it is then a convex hyperbolic constraint. By adding the auxiliary variable $t \in \mathbf{R}$, it can be written as one *second-order cone* (SOC) and two linear constraints:

$$\left\| \begin{bmatrix} 2\,t \\ \Sigma_{ii} - \Sigma_{jj} \end{bmatrix} \right\| \leq \Sigma_{ii} + \Sigma_{jj}, \quad \underline{\rho}_{ij} t \geq \Sigma_{ij} \geq \overline{\rho}_{ij} t.$$

4.4.3 Constraints on the variance of specific portfolios

Box constraints are a particular case of linear constraints on the covariance matrix. As an example of the use of other linear constraints, suppose we have better statistical information about the return of a given portfolio u (*e.g.*, the market portfolio). That is, we have a confidence interval for the variance of the portfolio return. We can then

define the linear constraint

$$\underline{s} \leq u^T \Sigma\, u \leq \overline{s},$$

where $\underline{s}$ and $\overline{s}$ are the lower and upper bounds on the variance of portfolio u. This is only useful if it results in a tighter specification for Σ than, say, the confidence intervals for its individual entries. Of course, constraints such as this can be included simultaneously for any number of portfolios.

4.4.4 Factor models

Factor models, unfortunately, are not easily handled. Consider a one factor model, that is, a diagonal plus rank one covariance matrix:

$$\Sigma = \mathbf{diag}(d) + bb^T.$$

In general, if the uncertainty is specified by a set which is convex in the d_i and b_i, $i = 1, \ldots, n$ (*e.g.*, a box constraint), a non-convex quadratic problem results. (Note that, in this case, we can parameterize Σ in the variables d_i, b_i, $i = 1, \ldots, n$, and dispense with the positive-semidefinite constraint.)

4.5 Solving the analysis problem

If one of the objectives (4.2), (4.3), or (4.4) given in §4.2 is coupled with any of the convex uncertainty sets described in §4.3 and §4.4, a convex program results. While nonlinear and non-differentiable, these programs have a global optimum, and can be efficiently solved. Thus, we can treat problems like: find the worst case downside risk given box uncertainty in Σ; or, find the worst case tracking error given box uncertainty in Σ and ellipsoidal uncertainty in μ. We will discuss two solution techniques: semidefinite programming, and projection methods.

4.5.1 Solution by semidefinite programming

Convex programming methods developed in recent years can efficiently solve all the convex problems previously described, even for large problem sizes. These methods are discussed, *e.g.*, in Vandenberghe and Boyd [68]. Currently available semidefinite programming software packages can handle problems of size over $n = 100$, on an inexpensive personal computer.

The computational complexity of these methods is provably polynomial in problem size. On the other hand, computing resources show no sign of departing from exponential growth and, as a consequence, within a few years semidefinite programs of size well into the thousands will be readily handled.

4.5.2 Solution by projection methods

There are alternative methods for the solution of the analysis problem that further exploit specific structure in the problem. These methods are based on the fact that it is easy to compute the projection on the sets defined by some of the constraints discussed above. Unlike in §4.5.1, this approach is not general, and depends on the particular problem under consideration. Its effectiveness relies on the projections on the objective level and constraint sets being computationally inexpensive.

For conciseness, we consider the example (4.1) given in the introduction, that is,

$$\begin{array}{ll} \text{maximize} & w^T \Sigma\, w \\ \text{subject to} & \underline{\Sigma}_{ij} \leq \Sigma_{ij} \leq \overline{\Sigma}_{ij}, \quad i,j = 1,\ldots,n \\ & \Sigma \succeq 0, \end{array} \tag{4.6}$$

where the variable is the symmetric matrix $\Sigma \in \mathbf{R}^{n\times n}$. This is a problem for which a projection method can be used, since all the required projections are readily computed.

Several optimization algorithms have been developed that are based on iterated projections (see, *e.g.*, Bertsekas [11]). In practice, we have found a variation of the projection arc algorithm to be effective – but, as noted before, with the caveat that our numerical experience is not extensive.

We next show how to compute projections onto objective function level sets, and onto the sets defined by the two constraints in the above problem: the box constraint and the positivity constraint. All projections are given in the Frobenius norm (consistency of norms is required to ensure the convergence of a projection based method.)

We also describe the problem dual to (4.6) which, for some problems, is more readily solved.

Projection on the objective level set

The gradient of the variance objective function (4.6) is a rank-one matrix,

$$\nabla_\Sigma(w^T\Sigma\, w) = ww^T.$$

Hence, the Euclidean projection (*i.e.*, for matrix 2-norm) on the objective level set $\mathcal{V} = \{\Sigma : w^T\Sigma\, w \geq \sigma^2\}$ is given by

$$P_{\text{obj}}(\Sigma) = \Sigma + \gamma ww^T, \qquad \text{where} \quad \gamma = \frac{\sigma^2 - w^T\Sigma w}{(w^Tw)^2}.$$

It can be shown that this also yields the Frobenius metric projection.

Projection on the box constraint set

The Frobenius norm projection on the set defined by the box constraint (4.6) is easily seen to be

$$[P_{\text{box}}(\Sigma)]_{ij} = \begin{cases} \underline{\Sigma}_{ij}, & \Sigma_{ij} < \underline{\Sigma}_{ij} \\ \overline{\Sigma}_{ij}, & \Sigma_{ij} < \overline{\Sigma}_{ij} \\ \Sigma_{ij}, & \text{otherwise.} \end{cases}$$

Projection on the positive semidefinite cone

The projection of Σ on the positive semidefinite cone (4.6) is also easily computed. This projection is obtained by computing the eigenvalue decomposition of Σ and

taking only the dyads with positive eigenvalues, *i.e.*,

$$P_{\text{psd}}(\Sigma) = U \operatorname{diag}\left((\lambda_i(\Sigma))_+ \right) U^T,$$

where $\Sigma = U \operatorname{diag}(\lambda_i(X)) U^T$ is the decomposition of Σ, and $(x)_+ = \max\{x, 0\}$. This projection is valid in both Euclidean and Frobenius metrics. (For proof, consider the problem of minimizing $\|X - \Sigma\|_{\text{F}}$ subject to $X \succeq 0$, and use the decomposition of Σ plus the invariance of the norm with respect to unitary transformations.)

Solution by Iterated projection

For a given value of the objective, σ^2, the problem can be determined to be feasible or infeasible by iterating the 3 projections, P_{obj}, P_{box}, and P_{psd}. The problem is feasible if the objective level set $\mathcal{V} = \{\Sigma : w^T \Sigma\, w \geq \sigma^2\}$ and the two sets defined by each of the problem constraints have non-empty intersection.

If the three sets do indeed have non-empty intersection (*i.e.*, if the optimal objective is higher than σ^2) the convexity of the sets and the fact that all projection are in the same (Frobenius) norm guarantees that the cycling of the projections will converge to a point in the intersection of the sets. Further, if the intersection of the sets is empty (*i.e.*, if the optimal objective is lower than σ^2), the cycling of the projections will converge to a fixed cycle.

This procedure can be repeated with bissection of σ^2, to find the optimal value of (4.6), which lies at the threshold between feasibility and infeasibility. We have then the following, simple algorithm:

1. Pick a value σ^2 for the objective, and cycle through the 3 projections: $S_1 = P_{\text{obj}}(\Sigma),\ S_2 = P_{\text{box}}(S_1),\ \Sigma = P_{\text{psd}}(S_2)$.

2. If σ^2 is less than the optimal value, step 1 will converge to a fixed point (Σ, S_1, S_2 equal to each other).
   If σ^2 is greater than the optimal value, step 1 will converge to a fixed cycle (Σ, S_1, S_2 repeating from cycle to cycle).

3. Repeat from step 1, with bisection on σ^2, to find the optimal value to the desired precision.

In practice, this is not an effective method. The convergence for the case when the objective value is feasible can be quite slow, which makes it hard to reliably detect infeasibility (since this detection relies on non-convergence to a point). Hence, what should be a lower bound for the bisection is easily mistaken for an upper bound, leading to erroneous results. A more effective method is presented next.

Solution by search along the arc of iterated projections

A more effective optimization method can be obtained by performing a line search along the gradient projection arc (see, *e.g.*, Bertsekas [11, §2.3]). The idea is, from a given starting point, to scale the gradient of the objective and project it on the constraint set. Bisection is then performed on the gradient scaling, to find the projected point with the best objective value. This is then used as a new starting point for the next iteration. The procedure is repeated until convergence.

However, in our case the projection arc cannot be found easily. We easily project on each of two convex constraints (box and positive semidefinite), but no easy way to compute the projection on the intersection of these two constraints. We can, nevertheless, use the two projections iteratively to find an arc of feasible points (which we may call a "pseudo-projection arc"). With this modification, the method can still be shown to converge. We have then the following algorithm:

1. Given the current point Σ and gradient scaling κ, let $S := \Sigma + \kappa \nabla_\Sigma (w^T \Sigma\, w) = \Sigma + \kappa\, w w^T$.

2. Repeat $S := P_{\text{psd}}(P_{\text{box}}(S))$ until S is feasible.

3. Repeat steps 1 and 2 as required to perform bisection search on κ to maximize $w^T S w$.

4. Update the current point to be the result of the bissection search, $\Sigma := S$, and repeat from step 1 (until convergence of Σ).

The dual problem

In some instances it may be more effective to solve the dual problem. This should be the case, in particular, if the dual problem includes fewer constraints and the projections are therefore easier to apply. The dual of (4.6) is

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(\overline{\Sigma}\,\overline{\Lambda} - \underline{\Sigma}\,\underline{\Lambda}) \\ \text{subject to} & Z + \overline{\Lambda} - \underline{\Lambda} = -ww^T \\ & \overline{\Lambda}_{ij} \geq 0, \quad \underline{\Lambda}_{ij} \geq 0, \quad Z \succeq 0. \end{array} \tag{4.7}$$

If (4.6) is feasible and bounded (*i.e.*, has a finite optimal objective), then (4.7) is also finite and bounded, and the optimal objectives of the two problems are identical [68].

Note that this can be equivalently written without the variable Z, using the constraint $\overline{\Lambda} - \underline{\Lambda} + ww^T \succeq 0$. Alternatively, by using a translation to make the box constraints symmetric and using $\Lambda = \overline{\Lambda} - \underline{\Lambda}$, the number of dual variables can be reduced. The dual problem (4.7) is then equivalent to

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}\left(\frac{\overline{\Sigma}+\underline{\Sigma}}{2} Z - \frac{\overline{\Sigma}-\underline{\Sigma}}{2} \Lambda\right) \\ \text{subject to} & \Lambda_{ij} \geq Z_{ij} + w_i w_j, \quad \Lambda_{ij} \geq -Z_{ij} - w_i w_j \\ & Z \succeq 0. \end{array} \tag{4.8}$$

Obvious extensions apply. For $\underline{\Sigma}_{ij} = \overline{\Sigma}_{ij}$ (*i.e.*, Σ_{ij} precisely known), the corresponding Λ_{ij} and the constraints associated with it are omitted. For Σ_{ij} unconstrained (*i.e.*, no knowledge about Σ_{ij}), we have that $\Lambda_{ij} = 0$, and the constraint $Z_{ij} = -w_i w_j$ is used.

4.6 The design problem

We now incorporate the previous analysis into portfolio optimization. The discussion so far has addressed the analysis problem, where the portfolio is fixed (and known). The corresponding design problem consists in selecting a portfolio using the worst-case risk as a criterion for the desirability of a particular portfolio. In the simplest

form of the problem, the goal is to find the portfolio that is optimal in the sense of the previous analysis, that is, the portfolio that minimizes the worst-case risk. The minimization is subject to constraints, of course, such as an upper bound on the budget, a lower bound on the expected return, or others.

For simplicity, we consider worst-case design for the classical mean-variance tradeoff problem, with uncertainty in the covariance matrix Σ. The same approach can be used for the other problems introduced in §4.2.

Consider then the problem of selecting the portfolio with lowest risk (defined as the worst-case portfolio variance), subject to a lower bound on expected return, and subject to budget and shorting constraints:

$$\begin{array}{ll} \text{minimize} & \max_{\Sigma \in \mathcal{S}} (w^T \Sigma w) \\ \text{subject to} & \mathbf{1}^T w = 1, \quad w_i \geq w_{\min}, \quad \mu^T w \geq R_{\min}, \end{array} \tag{4.9}$$

where $\mathcal{S} = \{ \Sigma \in \mathbf{R}^{n \times n} \mid \Sigma \succeq 0, \ \underline{\Sigma}_{ij} \leq \Sigma_{ij} \leq \overline{\Sigma}_{ij} \}$ is a convex compact set, so that the max is well defined. In practice, we will want to find the tradeoff curve between risk and return, which is obtained by solving this program ranging over different values of $R_{\min}$.

The uncertainty set can be scaled around some nominal point to make the design more "cautious" or more "aggressive". In the limit where the uncertainty set is reduced to a point (*i.e.*, to the nominal value of the mean and covariance of returns) the worst-case design reduces to the standard portfolio selection problem.

4.6.1 Min-max and max-min

Consider problem (4.9) in the form

$$\begin{array}{ccc} \min & \max & w^T \Sigma w, \\ w \in \mathcal{W} & \Sigma \in \mathcal{S} & \end{array} \tag{4.10}$$

where $\mathcal{W} = \{\, w \in \mathbf{R}^n \mid \mathbf{1}^T w = 1,\ w_i \geq w_{\min},\ \mu^T w \geq R_{\min}\}$ is a convex compact set, so that the min is well defined. This is equivalent to the problem

$$\begin{array}{lll} \min & & \max \quad \mathbf{Tr}(C\Sigma), \\ C \succeq ww^T,\ w \in \mathcal{W} & & \Sigma \in \mathcal{S} \end{array}$$

which has an inner product as objective function. We can now use the min-max theorem of game theory to switch the order of the min and max operators (see, *e.g.*, Bertsekas [11, §5.4.3], or Luenberger [40, §7.13]). The sets associated with both the min and max operators are convex. In particular, the newly added constraint is convex. Using the Schur complement, it is seen to be equivalent to

$$\left[\begin{array}{cc} C & w \\ w^T & 1 \end{array}\right] \succeq 0.$$

In the form stated the set associated with the min operator is not compact, as required by the theorem. However, since $\Sigma \succeq 0$ it is always possible to upper bound C without changing the solution of the problem, which makes the set compact (with the upper bound applied to, say, the spectral norm of C). Therefore, problem (4.10) is equivalent to

$$\begin{array}{lll} \max & & \min \quad w^T \Sigma\, w. \\ \Sigma \in \mathcal{S} & & w \in \mathcal{W} \end{array} \tag{4.11}$$

The design problem is convex overall, and we can equivalently solve it in either min-max or max-min form. In practice, the numerically most convenient form should be used.

4.7 Solving the design problem

In this section, we briefly indicate how the robust portfolio design problem can be effectively solved by analytic center cutting plane methods. For simplicity, the discussion here will focus on the variance problem. By using the results in Chapter 5, this approach is generalizable to the other robust portfolio problems. At the end of

the section we briefly indicate alternative solution methods.

Define the function $\phi : \mathbf{R}^n \to \mathbf{R}$,

$$\phi(w) = \max_{\Sigma \in \mathcal{S}} \; w^T \Sigma \, w,$$

and denote the corresponding optimal variable by

$$\Sigma^*(w) = \text{argmax}_{\Sigma \in \mathcal{S}} \; w^T \Sigma \, w.$$

For a convex $\mathcal{S}$, it is easily shown that the function

$$g(w) = 2 \, \Sigma^*(w) \, w$$

is a subgradient of $\phi(w)$. Hence, once $\phi(w)$ has been computed (*i.e.*, the associated convex program has been solved) the subgradient is obtained at essentially no extra computational cost. The subgradient defines a supporting hyperplane for the sublevel set $\{v : \phi(v) \leq \phi(w)\}$ (the plane orthogonal to $g(w)$ that passes through w).

With $\mathcal{S} = \{ \, \Sigma \in \mathbf{R}^{n \times n} \mid \Sigma \succeq 0, \; \underline{\Sigma}_{ij} \leq \Sigma_{ij} \leq \overline{\Sigma}_{ij} \}$ as before, problem (4.9) can be written

$$\begin{array}{ll} \text{minimize} & \phi(w) \\ \text{subject to} & w \in \mathcal{W}. \end{array}$$

Solving the mathematical program that computes $\phi(w)$ (and $\Sigma^*(w)$, $g(w)$) for a given w, provides an "oracle" for an analytic center cutting plane method, for which polynomial complexity has been established. For more on cutting plane methods see, *e.g.*, [32, 49, 28] (and the references therein.)

A method for solving this problem is as follows. Starting from any $w \in \mathcal{W}$:

1. Solve the inner problem: compute $\phi(w)$, a subgradient, and the corresponding supporting hyperplane.

2. Find the analytic center of the intersection of $\mathcal{W}$ with the half-planes defined by the previously computed supporting hyperplanes. Let w be the analytic center.

3. Repeat from 1, until w converges.

The overall complexity of this algorithm is polynomial in problem size. The cutting plane method is of polynomial complexity, and the same is true for a number of methods that solve the semidefinite program from which the cutting plane is obtained. Finding the analytic center requires the minimization of the logarithmic barrier function. In practice, the analytic centering problem does not have to be very precisely solved, which can save a significant amount of computational effort.

An alternative solution method exploits duality. The optimal value of the inner problem can equivalently be obtained by solving the associated dual problem. The min-max problem is then reduced to a minimization over all program variables. It is an open question whether this always results in a program that is easily handled by existing methods. In the case of (4.9), this approach results in an SDP, which is readily solved (Laurent El Ghaoui, personal communication.)

Another approach consists in developing self-concordant barrier functions for the min-max problems, allowing for the direct application of interior-point methods (Reha Tutuncu, personal communication.)

Chapter 5

Perturbation of conic programs

5.1 Introduction

We consider how changes in the parameters of a convex program translate into changes in the optimal value of the program objective. In particular, if the optimal value is convex with respect to the parameter of interest, we wish to obtain a subgradient. We will see that this can be written as a simple function of the optimal primal and dual variables. If the program is solved by, say, a primal-dual interior-point method, the optimal primal and dual variables are readily available. The subgradient can then be obained with virtually zero computational effort.

We will work in the quite general framework of convex conic programming over normed linear vector spaces. This allows for succinct derivations, while the results can be immediately specialized for more concrete optimization problems, such as semidefinite programming. These results can be applied to build algorithms for robust optimization problems, including the robust portfolio design problems discussed in Chapter 4.

5.2 Perturbation of the program parameters

5.2.1 Primal and dual programs, optimality conditions

The primal problem

Given normed linear vector spaces $\mathcal{Y}$ and $\mathcal{X}$, consider the convex programming problem

$$\begin{array}{ll} \text{minimize} & \langle Y, C\rangle \\ \text{subject to} & X \succeq_{\mathcal{K}} 0 \\ & X = \mathcal{A}(Y) + B \end{array} \tag{5.1}$$

where the problem variables are $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. The problem data are the convex cone (with nonempty interior) $\mathcal{K} \subset \mathcal{X}$, the linear operator $\mathcal{A} : \mathcal{Y} \longmapsto \mathcal{X}$, the vector $B \in \mathcal{X}$, and the linear functional $C \in \mathcal{Y}^*$. We refer to this as the *primal program*, and denote the optimal value of the objective of this program by ϕ^*.

The dual problem

A *dual program* associated with (5.1) is

$$\begin{array}{ll} \text{maximize} & -\langle B, Z\rangle \\ \text{subject to} & Z \succeq_{\mathcal{K}^*} 0, \\ & \mathcal{A}^*(Z) = C, \end{array} \tag{5.2}$$

where the problem variable is $Z \in \mathcal{X}^*$. $\mathcal{K}^*$ is the cone dual to $\mathcal{K}$,

$$\mathcal{K}^* \;=\; \{\; Z \mid \langle X, Z\rangle \geq 0,\; \forall\; X \in \mathcal{K} \;\}$$

(also assumed with non-empty interior), and $\mathcal{A}^*$ is the linear operator adjoint to $\mathcal{A}$,

$$\langle \mathcal{A}(Y), Z\rangle = \langle Y, \mathcal{A}^*(Z)\rangle\,,\; \forall\; Y \in \mathcal{Y},\; Z \in \mathcal{X}^*.$$

Here's a brief outline of the derivation of the dual. Define the Lagrangian

$$\mathcal{L}(Y, X, Z, V) = \langle Y, C\rangle - \langle X, Z\rangle + \langle X - \mathcal{A}(Y) - B, V\rangle ,$$

where $Z, V \in \mathcal{X}^*$. Given the dual function

$$g(Z, V) = \inf_{Y,X} \mathcal{L}(Y, X, Z, V),$$

$g(Z, V) > -\infty$ implies $Z = V$ and

$$\mathcal{A}^*(V) = C.$$

With these conditions,

$$g(Z, V) = g(Z) = -\langle B, Z\rangle .$$

From the Lagrangian it is easily shown that $g(Z) \leq \langle Y, C\rangle$ for any $Z \succeq_{\mathcal{K}^*} 0$, and that $\langle X, Z\rangle = 0$ implies $g(Z) = \langle Y, C\rangle$.

Optimality conditions

Optimality conditions for X and Z are given by the primal and dual constraints

$$X \succeq_{\mathcal{K}} 0, \quad X = \mathcal{A}(Y) + B, \quad Z \succeq_{\mathcal{K}^*} 0, \quad \mathcal{A}^*(Z) = C,$$

and by

$$\langle Y, C\rangle = -\langle B, Z\rangle ,$$

which is equivalent to

$$\langle X, Z\rangle = 0.$$

5.2.2 Perturbation of C

Consider the primal and dual programs, with the linear functional C replaced by $C + \Delta C$. The optimal variables for the perturbed primal and dual programs are denoted by $X + \Delta X$, $Y + \Delta Y$, and $Z + \Delta Z$ (where X, Y, and Z are the optimal

variables for the unperturbed programs). The optimality conditions are now

$$\begin{aligned}
&X + \Delta X \succeq_{\mathcal{K}} 0,\\
&X + \Delta X = \mathcal{A}(Y + \Delta Y) + B,\\
&Z + \Delta Z \succeq_{\mathcal{K}^*} 0,\\
&\mathcal{A}^*(Z + \Delta Z) = C + \Delta C,\\
&\langle Y + \Delta Y, C + \Delta C\rangle = -\langle B, Z + \Delta Z\rangle .
\end{aligned}$$

From this we derive

$$\begin{aligned}
\Delta X &= \mathcal{A}(\Delta Y),\\
\mathcal{A}^*(\Delta Z) &= \Delta C.
\end{aligned}$$

The change in the optimal objective value is

$$\begin{aligned}
\Delta\phi^* &= \langle Y + \Delta Y, C + \Delta C\rangle - \langle Y, C\rangle\\
&= -\langle B, Z + \Delta Z\rangle + \langle B, Z\rangle\\
&= -\langle B, \Delta Z\rangle\\
&= \langle \mathcal{A}(Y) - X, \Delta Z\rangle\\
&= \langle Y, \mathcal{A}^*(\Delta Z)\rangle - \langle X, \Delta Z\rangle\\
&= \langle Y, \Delta C\rangle - \langle X, Z + \Delta Z\rangle\\
&\leq \langle Y, \Delta C\rangle ,
\end{aligned}$$

where the inequality follows from X being in the primal cone and $Z + \Delta Z$ in the dual cone. Hence, a subgradient of $-\phi^*$ with respect to C is given by the linear functional defined by $-Y$.

5.2.3 Perturbation of B

Consider again the primal and dual programs, now with the vector B replaced by $B + \Delta B$. The optimal variables for the perturbed primal and dual programs are again

denoted by $X + \Delta X$, $Y + \Delta Y$, and $Z + \Delta Z$. The optimality conditions are now

$$\begin{aligned}
&X + \Delta X \succeq_{\mathcal{K}} 0, \\
&X + \Delta X = \mathcal{A}(Y + \Delta Y) + B + \Delta B, \\
&Z + \Delta Z \succeq_{\mathcal{K}^*} 0, \\
&\mathcal{A}^*(Z + \Delta Z) = C, \\
&\langle Y + \Delta Y, C \rangle = -\langle B + \Delta B, Z + \Delta Z \rangle .
\end{aligned}$$

From this we derive

$$\begin{aligned}
&\Delta X = \mathcal{A}(\Delta Y) + \Delta B, \\
&\mathcal{A}^*(\Delta Z) = 0.
\end{aligned}$$

The change in the optimal objective value is

$$\begin{aligned}
\Delta\phi^* &= \langle Y + \Delta Y, C \rangle - \langle Y, C \rangle \\
&= \langle \Delta Y, C \rangle \\
&= \langle \Delta Y, \mathcal{A}^*(Z) \rangle \\
&= \langle \Delta X - \Delta B, Z \rangle \\
&= \langle X + \Delta X, Z \rangle - \langle \Delta B, Z \rangle \\
&\geq -\langle \Delta B, Z \rangle ,
\end{aligned}$$

where the inequality follows from $X + \Delta X$ being in the primal cone and Z in the dual cone. Hence, a subgradient of ϕ^* with respect to B is given by the linear functional defined by $-Z$.

5.2.4 Perturbation of $\mathcal{A}$

Consider the primal and dual programs, with the linear operator $\mathcal{A}$ replaced by $\mathcal{A} + \Delta\mathcal{A}$. The optimal variables for the perturbed primal and dual programs are denoted

by $X + \Delta X$, $Y + \Delta Y$, and $Z + \Delta Z$. The optimality conditions are now

$$\begin{aligned}
&X + \Delta X \succeq_{\mathcal{K}} 0,\\
&X + \Delta X = (\mathcal{A} + \Delta\mathcal{A})(Y + \Delta Y) + B,\\
&Z + \Delta Z \succeq_{\mathcal{K}^*} 0,\\
&(\mathcal{A} + \Delta\mathcal{A})^*(Z + \Delta Z) = C,\\
&\langle Y + \Delta Y, C\rangle = -\langle B, Z + \Delta Z\rangle .
\end{aligned}$$

From this we derive

$$\begin{aligned}
&\Delta X = \mathcal{A}(\Delta Y) + \Delta\mathcal{A}(Y + \Delta Y),\\
&\mathcal{A}^*(\Delta Z) + \Delta\mathcal{A}^*(Z + \Delta Z) = 0.
\end{aligned}$$

The change in the optimal objective value is

$$\begin{aligned}
\Delta\phi^* &= \langle Y + \Delta Y, C\rangle - \langle Y, C\rangle\\
&= \langle \Delta Y, C\rangle\\
&= \langle \Delta Y, \mathcal{A}^*(Z)\rangle\\
&= \langle \mathcal{A}(\Delta Y), Z\rangle\\
&= \langle \Delta X - \Delta\mathcal{A}(Y + \Delta Y), Z\rangle\\
&= \langle \Delta X, Z\rangle - \langle \Delta Y, \Delta\mathcal{A}^*(Z)\rangle - \langle \Delta\mathcal{A}(Y), Z\rangle\\
&\geq -\langle \Delta Y, \Delta\mathcal{A}^*(Z)\rangle - \langle \Delta\mathcal{A}(Y), Z\rangle .
\end{aligned} \tag{5.3}$$

An alternative (dual) derivation yields

$$\begin{aligned}
\Delta\phi^* &= \langle Y + \Delta Y, C\rangle - \langle Y, C\rangle\\
&= -\langle B, \Delta Z\rangle\\
&= -\langle X - \mathcal{A}(Y), \Delta Z\rangle\\
&= -\langle X, \Delta Z\rangle + \langle Y, \mathcal{A}^*(\Delta Z)\rangle\\
&= -\langle X, \Delta Z\rangle - \langle Y, \Delta\mathcal{A}^*(\Delta Z)\rangle - \langle Y, \Delta\mathcal{A}^*(Z)\rangle\\
&\leq -\langle \Delta\mathcal{A}(Y), \Delta Z\rangle - \langle \Delta\mathcal{A}(Y), Z\rangle .
\end{aligned} \tag{5.4}$$

Assume that the optimal Y and Z are locally Lipschitz continuous in $\mathcal{A}$. Then, as $\|\Delta\mathcal{A}\| \to 0$, the first term in the right-hand-side of inequality (5.3) is $O(\|\Delta\mathcal{A}\|^2)$. The same is true for the first term in the right-hand-side of inequality (5.4). From

the opposite inequalities we conclude

$$d\phi^* = -\langle d\mathcal{A}(Y), Z\rangle .$$

This can also be written as

$$\frac{d\phi^*}{d\mathcal{A}}(\,\cdot\,) = -Z\ \langle Y, \cdot\,\rangle .$$

The derivative of the optimal primal cost with respect to the linear operator $\mathcal{A}$ is itself a linear operator, with same input and output spaces as $\mathcal{A}$. It is equal to minus the outer product of the optimal dual and primal variables, Z and Y.

Consider now the space of real vectors, with $Y = y \in \mathbf{R}^n$, $Z = z \in \mathbf{R}^m$, and the operator $\mathcal{A}$ specified by the matrix $A \in \mathbf{R}^{m\times n}$. The derivative is then

$$\frac{d\phi^*}{dA} = -z\, y^T.$$

(Assuming that the optimal y and z are Lipschitz continuous in A. We conjecture that if ϕ^* is (locally) convex in $\mathcal{A}$, the outer product should be a subgradient, even if Y or Z are not continuous.) Note that this result holds for any conic convex program, including LP, SOCP, and SDP.

Chapter 6

Conclusions and directions for future work

Large-scale optimization has many areas of application in finance. A unifying idea in this thesis is to effectively address a number of important problems in these areas by exploiting new efficient interior-point methods for nonlinear convex programming. While such methods are of polynomial complexity (in simple implementations, cubic) in problem dimension, the availability of computing resources over time shows no signs of departing from geometric growth. As a consequence, interior-point methods will be able to handle very large problems, in very short run-times in the near future. Currently, run-times are in the order of a minute for portfolio selection problems with a few hundred variables, on an inexpensive personal computer, using generic software that is not optimized for portfolio problems.

We included a more detailed presentation of second-order cone programming, a problem class that lies between linear (or quadratic) programming and semidefinite programming. Like LP and SDP, SOCPs can be solved very efficiently by primal-dual interior-point methods (and in particular, far more efficiently than by treating the SOCP as an SDP). Moreover, a wide variety of practical problems can be formulated as second-order cone problems.

Perhaps the most important direction for future work in SOCP lies in algorithms that exploit sparsity or some other problem structure – including those that arise in

portfolio optimization problems.

In Chapter 3 we described a number of portfolio optimization problems that are convex, and therefore efficiently solved. If fixed transaction costs are included, the resulting problem is not convex. In this case, we showed how to compute a global upper bound from a convex relaxation, and proposed a heuristic for computing an approximate solution (which yields, of course, a lower bound). Computational experiments suggest that the gap between our heuristic suboptimal solution and our guaranteed upper bound is very often small. If further accuracy is needed, the upper bound and the heuristic method can be incorporated in a branch-and-bound method.

Much work needs to be done in this area, including a treatment of how well the heuristic presented here performs in practice. This can be investigated, for example, by using a global branch-and-bound algorithm to compute the global optimum, and comparing the results with those of the heuristic. Also, many extensions of the portfolio selection problem can be considered, including for multiperiod problems.

In Chapter 4 we saw another application of convex programming, the worst-case risk analysis of a portfolio. For many cases of interest, computing the maximum risk of a portfolio given uncertainty in the means and covariances of asset returns is a semidefinite programming problem. Its global solution can be efficiently computed, even for large problems. While not as general, this approach is more accurate and much faster than Monte Carlo methods. The computational effort required grows gracefully, which allows very large problems to be handled.

Solving portfolio optimization problems with great precision, when the problem parameters are inherently uncertain, is not a reasonable proposition. By using cutting plane methods, the worst-case risk analysis can be incorporated into portfolio selection, which allows "robust" portfolios to be designed. The sensitivity analysis of convex-conic programs, as discussed in Chapter 5, can be used in the construction of such cutting plane methods.

As noted in the introduction, the ideas in Chapters 4 and 5 are not thoroughly explored, and this is especially true in regards to numerical experience. The approach needs to be extensively tested on real data, and a statististical study is required to ascertain what models of uncertainty are appropriate for the covariance matrices.

Another major direction for future work will be the design and implementation of effective methods for robust portfolio synthesis. And the final test, of course, will be that of actual usage by practitioners in the field.

Bibliography

[1] I. Adler and F. Alizadeh. Primal-dual interior point algorithms for convex quadratically constrained and semidefinite optimization problems. Technical Report RRR 46-95, RUTCOR, Rutgers University, 1995.

[2] F. Alizadeh, J. P. Haeberly, M. V. Nayakkankuppam, and M. L. Overton. SDPPACK *User's Guide, Version 0.8 Beta.* NYU, June 1997.

[3] F. Alizadeh, J. P. Haeberly, M. V. Nayakkankuppam, M. L. Overton, and S. Schmieta. SDPPACK *User's Guide, Version 0.9 Beta.* NYU, June 1997.

[4] F. Alizadeh and S. H. Schmieta. Optimization with semidefinite, quadratic and linear constraints. Technical report, RUTCOR, Rutgers University, November 1997.

[5] E. Andersen. *MOSEK v1.0b User's manual*, 1999. Available from the url `http://www.mosek.com`.

[6] E. Andersen and K. Andersen. *APOS user's manual for QCOPT ver 1.00 Beta*. EKA consulting, May 1997. `http://www.samnet.ou.dk/~eda/`.

[7] K. Andersen. An efficient Newton barrier method for minimizing a sum of Euclidean norms. *SIAM J. on Optimization*, 6(1):74–95, 1996.

[8] K. Andersen, E. Christiansen, and M. Overton. Computing limits loads by minimizing a sum of norms. Technical Report 30, Institut for Matematik og Datalogi, Odense Universitet, September 1994.

[9] A. Ben-Tal and A. Nemirovski. Robust convex optimization. Technical report, Faculty of Industrial Engineering and Management, Technion, December 1996.

[10] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[11] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995.

[12] B. Blog, G. V. D. Hoek, A. H. G. Rinnooy Kan, and G. T. Timmer. The optimal selection of small portfolios. *Management Science.*, 29(7):792–798, July 1983.

[13] S. Boyd, C. Crusius, and A. Hansson. Control applications of nonlinear convex programming. *Journal of Process Control*, 8(5-6):313–324, 1998. Special issue for papers presented at the 1997 IFAC Conference on Advanced Process Control, June 1997, Banff.

[14] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[15] S. Boyd and L. Vandenberghe. Introduction to convex optimization with engineering applications. Course Notes, 1997. http://www.stanford.edu/class/ee364/.

[16] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. Technical Report SCCM-95-10, Stanford University. Computer Science Department, 1995.

[17] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed. Efficient algorithms for least square type problems with bounded uncertainties. Technical Report SCCM-96-16, Stanford University, 1996.

[18] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed. Parameter estimation in the presence of bounded data uncertainties. *SIAM J. on Matrix Analysis and Applications*, 1998. To appear.

[19] A. R. Conn and M. L. Overton. Primal-dual interior-point method for minimizing a sum of Euclidean distances. Draft, 1994.

[20] J. Czyzyk, S. Mehrotra, and S. J. Wright. *PCx User Guide*. Optimization Technology Cneter, March 1997.

[21] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[22] A. H. Delaney and Y. Bresler. Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography. *IEEE Transactions on Signal Processing*, 7(2):204–221, February 1998.

[23] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming*. Kluwer, 1993.

[24] D. R. Dreyer and M. L. Overton. Two heuristics for the Steiner tree problem. Technical Report 724, New York University Computer Science Department, 1996.

[25] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. on Matrix Analysis and Applications*, 18(4):1035–1064, October 1997.

[26] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. on Optimization*, 9(1):33–52, 1998.

[27] A. Fiacco and G. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. Wiley, 1968. Reprinted 1990 in the SIAM Classics in Applied Mathematics series.

[28] J.-L. Goffin and J.-P. Vial. Shallow, deep and very deep cuts in the analytic center cutting plane method. *Mathematical Programming*, 84(1):89–103, January 1999.

[29] D. Goldfarb, S. Liu, and S. Wang. A logarithmic barrier function algorithm for convex quadratically constrained quadratic programming. *SIAM J. on Optimization*, 1:252–267, 1991.

[30] X. Ju and N. D. Pearson. Using value-at-risk to control risk taking: how wrong can you be? *Journal of Risk*, 1(2):5–36, Winter 1999.

[31] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[32] J. E. Kelley. The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math*, 8(4):703–712, December 1960.

[33] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14:699–719, 1966.

[34] M. L. Leibowitz, L. N. Bader, and S Kogelman. *Return targets and shortfall risks: studies in strategic asset allocation*. Salomon Brothers, Irwin publishing, Chicago, 1996.

[35] John Lintner. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, 47(1):13–37, February 1965.

[36] M. S. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs and bounds on risk. Working paper, `http://www.stanford.edu/~boyd`, March 1999.

[37] M. S. Lobo, L. Vandenberghe, and S. Boyd. SOCP*: Software for Second-Order Cone Programming*. Information Systems Laboratory, Stanford University, 1997.

[38] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Applications*, 284(1-3):193–228, November 1998.

[39] A. Lucas and P. Klaasen. Extreme returns, downside risk, and optimal asset allocation. *The Journal of Portfolio Management.*, pages 71–79, Fall 1998.

[40] D. G. Luenberger. *Optimization By Vector Space Methods*. John Wiley & Sons, New York, N. Y., 1969.

[41] D. G. Luenberger. *Investment Science*. Oxford University Press, New York, 1998.

[42] Joseph Moncure March and Art Spiegelman. *The Wild Party: The Lost Classic by Joseph Moncure March; Drawings by Art Spiegelman*. Pantheon Books, Random House, 1928, 1994.

[43] H. M. Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

[44] H. M. Markowitz. *Portfolio Selection*. J. Wiley & Sons, New York, 1959.

[45] R. R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *Journal of computer and system sciences*, 12:108–121, 1974.

[46] J. P. Morgan and Reuters. Riskmetrics – technical document. Technical report, December 1996. Fourth Edition.

[47] Robb J. Muirhead. *Aspects of Multivariate Statistical Theory*. Wiley, 1982.

[48] A. Nemirovskii and K. Scheinberg. Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems. *Math. Programming*, 72(3, Ser. A):273–289, 1996.

[49] Y. Nesterov. Complexity estimates of some cutting plane methods based on the analytic barrier. *Mathematical Programming*, 69(1):149–176, July 1995.

[50] Yu. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[51] Yu. E. Nesterov and M. J. Todd. Self-scaled cones and interior-point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[52] Yu. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8(2):324–364, May 1998.

[53] M. H. Patel. Spherical minimax location problem using the Euclidean norm: formulation and optimization. *Computational Optimization and Applications*, 4:79–90, 1995.

[54] N. R. Patel and M. G. Subrahmanyam. A simple algorithm for optimal portfolio selection with fixed transaction costs. *Management Science.*, 28(3):303–314, 1982.

[55] A. F. Perold. Large-scale portfolio optimization. *Management Science.*, 30(10):1143–1160, October 1984.

[56] A. D. Roy. Safety first and the holding of assets. *Econometrica*, 20:413–449, 1952.

[57] M. Rudolph. *Algorithms for portfolio optimization and portfolio insurance*. Haupt, Switzerland, 1994.

[58] A. H. Sayed, A. Garulli, V. H. Nascimento, and S. Chandrasekaran. Exponentially-weighted iterative solutions for worst-case parameter estimation. In *Proceedings of the 5th IEEE Med. Conf. Control and Systems*, Cyprus, July 1997.

[59] Jason B. Schattman. *Portfolio selection under nonconvex transaction costs*. PhD thesis, Rutgers University, 2000.

[60] Katya Scheinberg. Parametric linear semidefinite programming. In *Handbook on Semidefinite Programming*, chapter 4.2, pages 93–110. Kluwer, 2000.

[61] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience series in discrete mathematics. John Wiley & Sons, 1986.

[62] William F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, 19(3):425–442, September 1964.

[63] J. Sturm. SeDuMi *Version 1.02*. McMaster University, 1998.

[64] L. G. Telser. Safety first and hedging. *Review of Economics and Statistics*, 23:1–16, 1955.

[65] T. Tsuchiya. A polynomial primal-dual path-following algorithm for second-order cone programming. Technical report, The Institute of Statistical Mathematics, Tokyo, Japan, October 1997.

[66] Reha Tütüncü. Personal communication, 2000.

[67] L. Vandenberghe and S. Boyd. SP*: Software for Semidefinite Programming. User's Guide, Beta Version.* Stanford University, October 1994. Available at `www.stanford.edu/~boyd`.

[68] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.

[69] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM J. on Matrix Analysis and Applications*, 19(2):499–533, April 1998.

[70] R. J. Vanderbei. LOQO user's manual. Technical Report SOL 92–05, Dept. of Civil Engineering and Operations Research, Princeton University, Princeton, NJ 08544, USA, 1992.

[71] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer Academic Publishers, 1997.

[72] P. Whittle. *Optimization under Constraints. Theory and Applications of Nonlinear Programming*. Wiley-Interscience, 1971.

[73] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.

[74] G. Xue and Y. Ye. An efficient algorithm for minimizing a sum of Euclidean norms with applications. *SIAM J. on Optimization*, 1996. To appear.

[75] Y. Ye. An $O(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50:239–258, 1991.

[76] Y. Zhang. *User's guide to LIPSOL: a matlab toolkit for linear programming interior-point solvers*. Math. & Stat. Dept., Univ. Md. Baltimore County, October 1994. Beta release.