Portfolio Optimization via Downside-Risk Aversion Model

by

Yuen-Lam Cheung

A research paper
presented to the University of Waterloo
in fulfillment of the
requirement for the course-work degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2009

Abstract

The mean-variance model has been enjoying high popularity among investors as a crucial portfolio construction tool. Unfortunately, there are a number of theoretical drawbacks that the model suffers from; one commonly criticized aspect is that variance is hardly adequate as a risk measure and does not distinguish downside risk from upside uncertainty. An expected-utility model using S-shaped utility function to strike a balance between the twin goals of minimizing risk and maximizing return has been gaining attention as an alternative option that provides a more realistic description of investors' risk perception. This exposition looks into the Mathematical groundwork for the problem formulation and describes two major methods for solving the resultant optimization problem: a variant of active set method and a smoothing technique.

Acknowledgments

I would like to express my deep thanks to my supervisor, Professor Michael J. Best, for his infinite patience, useful education and continuing support throughout this year.

I would also like to express my gratitude to Professor Levent Tunçel and Professor Henry Wolkowicz for not only their useful comments on this essay but also their invaluable and continuing guidance.

Thanks to all professors and friends in the Faculty of Mathematics, who have opened my eyes to a wider world. Special thanks to Cristiane, Marcel and Mario, who have made my time here much easier and much more pleasant to get through.

I feel most indebted to my family, without whose unconditional support I would not have had the courage to pursue my dreams.

Last but definitely not least, many thanks go to all my friends — wherever you are — who have been by my side all these years, despite the fact that I am not even physically around most of the time. Without all your care, I would not have made it through this eventful year, and it would not have been possible for me to come to this point of accomplishing one milestone of life.

Contents

Chapter 1

Introduction

1.1 Outline

The celebrated portfolio optimization problem — allocation of wealth to different investment instruments in order to achieve the often conflicting twin goals of maximizing expected returns and minimizing exposed risk — received its first Mathematical treatment in 1950s from Markowitz [15]. Since then, variance of returns of available instruments has been widely used as a risk measure, reducing the portfolio optimization problem to a linearly constrained quadratic program:

$$\begin{aligned} \max_x \ & \mu^T x - \frac{1}{t} x^T \Sigma x \\ \text{s.t. } & Ax \leq b \end{aligned}$$

where μ is the vector of expected returns of all available instruments, Σ the covariance matrix, the inequality system $Ax \leq b$ collecting some practical constraints such as budget constraint, and $1/t$ (with $t > 0$) the risk aversion parameter.

On top of the simplicity of modeling, one reason for the popularity of the mean-variance approach is that the resultant quadratic program can be solved quickly by existing algorithms, for instance active set methods and more recently interior-point methods. Nonetheless, the simplicity of the mean-variance approach also causes some major drawbacks to the modeling of the portfolio optimization problem. One of the shortfalls is that gain and loss of the same magnitude are viewed as equally risky, which is empirically not true (see, for example, [12]), and this is not accounted for by a risk measure like variance which is symmetric in nature.

To emphasize the importance of downside risk — the undesirable uncertainty of having portfolio returns below an acceptable threshold — in the choice of portfolio, some alternative models have been proposed over the years. An increasingly popular model is to view portfolio optimization problem as a expected-utility maximization problem: the utility function is used to take into account both the returns of a portfolio and the risk associated with it. In fact, mean-variance model is a special instance of expected-utility maximization problem. It has been proposed that an S-shaped utility function with suitable properties could more accurately describe the risk perception. Interestingly, the problem of minimizing conditional value-at-risk, whose proxy has a piecewise linear objective function [20], falls into the class of optimization problem.

While this reference-dependent utility model might be closer to reality, the equivalent minimization problem becomes a non-convex one. To be able to apply the model in practice, efficient algorithms for this class of optimization problems are necessary. On top of the non-convexity of the objective function, another issue is the size of problem: the expected utility is often taken to be a weighted average of huge number of scenarios, which could be drawn from historical data, for example. Several different classes of techniques are available to deal with these aspects; two mainstream methods will be introduced in this essay.

The essay is organized as follows. Chapter 2 explains the derivation of the optimization problem of interest. Chapter 3 describes an active set method for solving the optimization problem. Chapter 4 outlines a smoothing technique as an alternative.

1.2 Notations

$\mathbb{R}_+$ stands for the set of non-negative real numbers, and $\mathbb{R}_{++}$ stands for the set of positive real numbers.

All vectors in this essay are column vectors. $\mathbb{1}$ stands for the vector of all ones (and its dimension is implied from the context). Entries of a vector $y \in \mathbb{R}^n$ are indexed by subscript in the form y_i ($i = 1, \dots, n$). For any n-vectors a and b, $a \geq b$ and $a > b$ respectively mean $a_i \geq b_i$ and $a_i > b_i$ for $i = 1, \dots, n$.

For any real number t, $[t]^+ := \max\{0, t\}$ and $[t]^- := \max\{0, -t\}$ denote the positive and negative parts of t respectively. For any vector $z = (z_1, \dots, z_n)^T \in \mathbb{R}^n$, the n-vectors $[z]^+$ and $[z]^-$ are defined such that the i-th entry of the vector $[z]^+ \in \mathbb{R}^n$ is given by $[z_i]^+$, and similarly the i-th entry of $[z]^- \in \mathbb{R}^n$ is given by $[z_i]^-$.

Iterates in an algorithm are indexed by superscript. For instance x^j may refer to the j-th iterate generated by an algorithm.

Chapter 2

Portfolio Optimization: A New Perspective

2.1 Background

The portfolio optimization problem is classically posed as follows: given one unit of wealth, what is the optimal proportion of investment (which is called *portfolio*) in different assets such that the total return is maximized and exposed risk is minimized? Several difficulties in answering this question arise. First of all, these two goals are often in conflict with one another. Moreover, risk is a subjective concept: quite often two different persons may have different risk perception towards the same portfolio. Finally, how should an abstract quality as risk be quantified?

In the framework of single-period portfolio selection problem, Markowitz [13][14] proposed to use the variance of the portfolio as a risk measure: in a universe of n assets, the vector $r \in \mathbb{R}^n$ of returns at the end of the period is unknown to the investor at the time of decision, so r is assumed to be a random vector (following some probability distribution P); the expected return and variance of a fixed portfolio $x \in \mathbb{R}^n$ are given respectively by $\mu^T x$ and $x^T \Sigma x$, where $\mu := \mathbb{E}^P(r)$ is the vector of expected returns and $\Sigma := [\mathbb{E}^P\big((r_i - \mu_i)(r_j - \mu_j)\big)]_{ij}$ is the covariance matrix. With risk aversion parameter $1/t$ (where $t > 0$), the risk of the portfolio x is taken as $(1/t)x^T \Sigma x$. The celebrated mean-variance model for portfolio choice is to maximize the quantity $\mu^T x - \frac{1}{t}\, x^T \Sigma x$ under certain constraints (such as bound constraints on the holding of each asset and the budget constraint). The constraints are assumed to be linear, and are collectively represented in the linear inequality system $Ax \leq b$ for some matrix A and vector b. Therefore the mean-variance model can be posed as follows:

$$\begin{aligned} \max_x \; & \mu^T x - \frac{1}{t}\, x^T \Sigma x \\ \text{s.t. } & Ax \leq b \end{aligned} \tag{2.1}$$

This model falls under the more general framework of utility maximization model: suppose an agent would like to maximize his expected utility $f(\cdot)$, which depends solely on the return of his portfolio, by choosing an optimal portfolio under the same set of constraints as above. Mathematically, he is solving the optimization problem

$$\begin{aligned} \max_x \; & \mathbb{E}^P\big(f(r^T x)\big) \\ \text{s.t. } & Ax \leq b \end{aligned}$$

If his utility function happens to be $f(r^T x) = r^T x - (1/t)(r^T x - \mu^T x)^2$, then

$$\begin{aligned} \mathbb{E}^P\big(r^T x - (1/t)(r^T x - \mu^T x)^2\big) &= \mu^T x - \frac{1}{t} \mathbb{E}^P \left(\sum_{i,j=1,\ldots,n} (r_i - \mu_i)(r_j - \mu_j) x_i x_j \right) \\ &= \mu^T x - \frac{1}{t} \sum_{i,j=1,\ldots,n} \Sigma_{ij} x_i x_j \\ &= \mu^T x - \frac{1}{t}\, x^T \Sigma x \end{aligned}$$

which is the same as the objective function in the mean-variance model.

In the quest of a risk measure that conforms to realistic risk perception in a better way, a growing number of literature (see [1], [3], [7], [8], [11], [22] and references therein) supports the view that a realistic utility function should display two features that have been omitted in neo-classical utility theory. First, an individual's utility is *reference-dependent* in the sense that marginal utility is decreasing only when a threshold (of portfolio returns in this case) is surpassed. Second, downside return (a return that is below the threshold) is more seriously penalized than upside return of the same magnitude (with respect to the threshold).

Consequently, an S-shaped utility function, rather than neo-classical utility functions (such as power utility functions which unconditionally display diminishing marginal utility) or quadratic utility function (as in the case of mean-variance analysis), should be used in the optimization problem (2.1). The underlying idea of an S-shaped, or rather convex-concave, utility function is that, in a given scenario, there is a threshold (which corresponds to the deflection point of the utility function) around which an individual is more sensitive to small changes, which means that incremental perceived risk is greater when that change is near that threshold. Moreover, the S-shaped utility function is *not* symmetric, in order to reflect the asymmetric penalization of downside risk. (See Figure 2.1.)

Assume that the agent's constraints are given by linear inequalities; the set of feasible portfolios is given by

$$R = \{x \in \mathbb{R}^n : Ax \leq b\}$$

where A is an $m \times n$ matrix and b is an m-vector. This may include common trade restrictions such as budget constraint (that is, $\mathbb{1}^T x = 1$), no-short-sell constraints (that is, $x \geq 0$) or some other bound constraints. Because of these restrictions, R is normally a bounded set.

2.2 Generic downside-risk averse utility function

Generic reference-dependent utility function[1] of the agent is defined as a linear translation of the function

[1]Reference-dependent utility function is introduced by Kahneman and Tversky as one of the central ingredients of prospect theory ([12], currently the most cited paper in Econometrica). In their paper the authors called this *value function*.

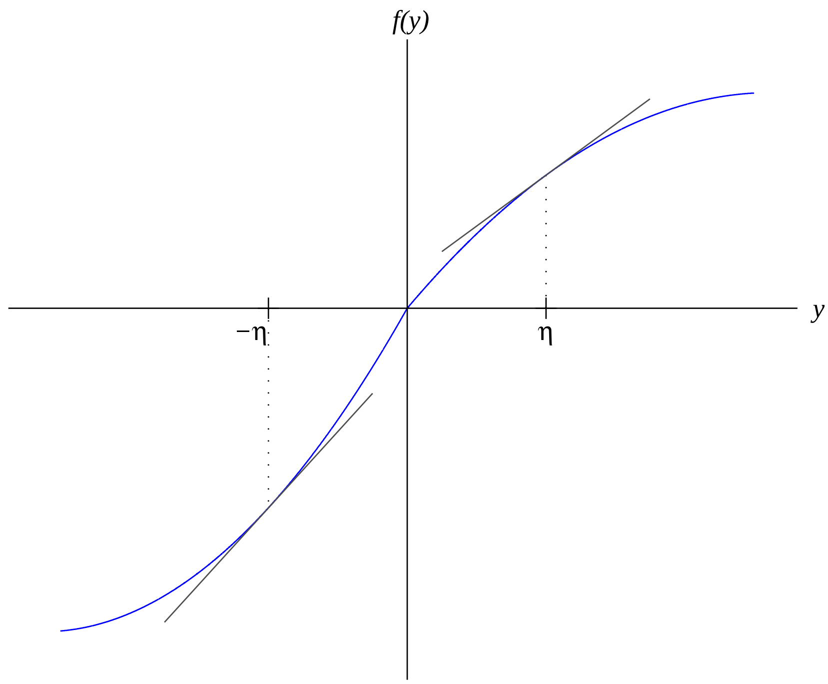

Figure 2.1: S-shaped utility function.

$$f(t) := \begin{cases} f^+(t) & \text{if } t \geq 0 \\ -f^-(-t) & \text{if } t \leq 0 \end{cases} \tag{2.2}$$

$$= f^+\big([t]^+\big) - f^-\big([t]^-\big). \tag{2.3}$$

where $f^+, f^- : \mathbb{R}_+ \to \mathbb{R}$ satisfy the following properties:

(P1) Monotonic utility and diminishing margin: $f^+, f^- : \mathbb{R}_+ \to \mathbb{R}$ are concave and strictly increasing.

(P2) Continuity at kink point: $f^+(0) = 0 = f^-(0)$.

(P3) Downside-risk aversion property: $f(y) + f(-y) < f(x) + f(-x)$ whenever $y > x > 0$.

Zero is taken as the reference point for this utility function, and $f^+, f^- : \mathbb{R}_+ \to \mathbb{R}$ correspond to the utility of the agent derived from "gain" and "loss" (with respect to the reference point). The mathematical expression of downside-risk aversion property is adopted from [12], depicting the attitude that a wider "symmetric" difference from the reference point leads to a lower overall utility level. Observe that f so defined is a convex-concave function, with zero being the inflection point.

If, in particular, f^+, f^- are twice differentiable on $\mathbb{R}_{++}$, then

(1) $(f^+)'$, $(f^-)' > 0$ and $(f^+)'', (f^-)'' \leq 0$ on $\mathbb{R}_{++}$; and

(2) $(f^+)'(y) \leq (f^-)'(y)$ for all $y > 0$. (In particular, the reference point zero attains the largest slope of f.)

2.3 Portfolio optimization via downside-risk aversion

With the generic utility function defined, the next step is to put (2.1) in a tractable form. Since the probability distribution P is not known in general, the expected utility is computed as follows: suppose there are S possible *states* or scenarios (which could come from historical data or Monte Carlo simulations, for example). A weight π_s is given to each state s, and the return vector of all assets at state s is given by r_s (to avoid peculiarity, assume that $r_s \neq 0$ for all s). The utility function of the agent at state s is given by $f_s : \mathbb{R} \to \mathbb{R}$ with reference point $\hat{y}_s$ is defined via some functions f_s^+, $f_s^- : \mathbb{R}_+ \to \mathbb{R}$ that satisfy (P1) to (P3):

$$f_s(y_s) = \begin{cases} f_s^+\big(y_s - \hat{y}_s\big) + c_s & \text{if } y_s \geq \hat{y}_s \\ -f_s^-\big(-(y_s - \hat{y}_s)\big) + c_s & \text{if } y_s < \hat{y}_s \end{cases} \tag{2.4}$$

$$= f_s^+\big([y_s - \hat{y}_s]^+\big) - f_s^-\big([y_s - \hat{y}_s]^-\big) + c_s \quad , \tag{2.5}$$

where the constant c_s is the level of utility at the reference point at state s.

The expected utility is simply the weighted sum of utility functions of all states; the canonical portfolio optimization problem can be posed as

$$\max_x \sum_{s=1}^{S} \pi_s f_s(r_s^T x) \quad \text{s.t.} \quad Ax \leq b \ . \tag{2.6}$$

There are several ways of solving (2.6). One mainstream strategy is to introduce new variables to convert the problem into an equivalent convex program; in the special case where the utility function is piecewise linear (or quadratic), it is possible to obtain an equivalent linear program (respectively quadratic program). Subsequently it is possible to employ existing general-purpose LP/QP algorithms (for example active set methods, interior point methods) to solve the lifted problem (that is, a new problem that is equivalent to the original one and that is obtained by introducing some more new variables — in this sense the original problem is lifted to a higher dimension), or to employ some specialized algorithms that circumvent the difficulty of dealing with the excessive amount of variables resulting from the lifting, by exploiting the structure of the new problem.

2.4 Conditional value at risk: an incidentally similar concept

Incidentally, minimization of conditional value-at-risk (CVaR), an investment decision problem related to portfolio optimization, can be formulated as a special instance of (2.6). CVaR is a risk measure that is gaining more and more attention due to its nice theoretical and mathematical characteristics. Specifically, CVaR is a coherent risk measure and, if the loss as a function of portfolio is convex, CVaR is a convex function of portfolio [17]. Consequently, minimization of CVaR could indeed be formulated as a convex program problem, provided that a convex loss function is chosen and that the feasible region is convex [20]. Loosely speaking, CVaR minimization problem can be viewed as an expected-utility maximization problem, with an appropriate choice of utility function. What is interesting is that the formulation of CVaR minimization problem happens to be a special case of (2.6).

The definition of CVaR is, as the name implies, conditional with respect to value at risk (VaR). Some terminology is needed for the definitions of VaR and CVaR. Let $f(x,y)$ (the *loss function*) be the loss of the portfolio $x \in \mathbb{R}^n$ associated with random vector $y \in \mathbb{R}^d$, which accounts for uncertainty in the model (therefore $f(x,\cdot)$ is a random variable too). y is assumed to follow a certain probability distribution characterized by probability density function $p(\cdot)$. For a given portfolio, the probability that the loss does not exceed a threshold α is given by

$$\Phi(x,\alpha) := \int_{f(x,y)\leq\alpha} p(y)dy \quad .$$

To ensure the subsequent mathematical concepts make sense, Φ is assumed to be everywhere continuous with respect to the second argument.

The β-VaR associated with a portfolio x is defined to be the lowest threshold $\alpha_\beta(x)$ such that with a probability β, the portfolio would not incur loss above $\alpha_\beta(x)$:

$$\alpha_\beta(x) := \inf\{\alpha \in \mathbb{R} : \Phi(x,\alpha) \geq \beta\} \tag{2.7}$$

Note that since $\Phi(x,\cdot)$ is assumed to be continuous for every x, the level set $\{\alpha \in \mathbb{R} : \Phi(x,\alpha) \geq \beta\}$ is closed and the infimum is attained if it is finite.

β-VaR can be thought of as a mark that signifies the undesirable $(1-\beta)$-quantile corresponding to "unacceptably" huge loss, so to speak. This mark itself, however, does not provide any information about *how* huge the loss could be — on average — beyond this unpleasant threshold, and this could render VaR a rather weak risk measure.

Therefore, β-CVaR, or expected shortfall, is introduced to provide additional information: if an unlucky investor lands on that $(1-\beta)$-quantile, on average how much would he lose? Mathematically, CVaR is defined by

$$\phi_\beta(x) := \frac{1}{1-\beta}\int_{f(x,y)\geq\alpha_\beta(x)} f(x,y)p(y)dy \ .$$

As can be seen, CVaR is simply the conditional expectation of loss given that the loss is above VaR. It has been shown [20] that β-CVaR can be computed by solving a convex program:

$$\phi_\beta(x) = \min_{\alpha\in\mathbb{R}} F_\beta(x,\alpha) \ , \tag{2.8}$$

where

$$F_\beta(x,\alpha) := \alpha + \frac{1}{1-\beta}\int_{f(x,y)\geq\alpha_\beta(x)} [f(x,y)-\alpha]^+ p(y)dy \tag{2.9}$$

is convex and continuously differentiable as a function of α.

Even more surprisingly, the problem of minimizing CVaR has a nice equivalent form:

$$\min_{x\in R}\phi_\beta(x) = \min_{(x,\alpha)\in R\times\mathbb{R}} F_\beta(x,\alpha) \quad . \tag{2.10}$$

$(\bar{x},\bar{\alpha})$ solves the problem on the right-hand side if and only if $\bar{x}$ solves the problem on the left-hand side and $\bar{\alpha}$ attains the CVaR $\phi_\beta(\bar{x})$. Here the set of feasible portfolio is again assumed to be given by $R=\{x : Ax\leq b\}$.

The integral in (2.9) can be approximated most easily by a finite collection of samples from the probability distribution: suppose $y_1,\ldots,y_S$ are randomly drawn samples from the probability distribution, then (2.9) could be replaced by

$$\tilde{F}_\beta(x,\alpha) := \alpha + \frac{1}{(1-\beta)S}\sum_{s=1}^{S}[f(x,y_s)-\alpha]^+ \ , \tag{2.11}$$

which is a convex piecewise linear function of α, as a proxy in the computation of CVaR in (2.8).

Assuming that all the uncertainty is represented by the vector of returns, whose probability distribution is approximated by some samples r_s ($s = 1, \ldots, S$), consider the case when the loss function is simply given by $(x, r) \mapsto -r^T x$, the negative value of return of the portfolio. Taking portfolio constraints into account, the minimization of CVaR formulated in (2.10) reduces to solving the following non-smooth convex program:

$$\min_{(x,\alpha)} \ \alpha + \frac{1}{(1-\beta)S}\sum_{s=1}^{S}[-r_s^T x-\alpha]^+ \quad \text{s.t.} \quad Ax \leq b \ . \tag{2.12}$$

The objective function as a sum of α and piecewise linear function of $(r;1)^T(x;\alpha)$ is again of the form (2.5); consequently, methods described in subsequent chapters can solve CVaR minimization problem too.

Chapter 3

Lifting and Active Set Method for Piecewise Linear Case

3.1 The lifted problem

This chapter outlines an active set method for solving the canonical utility maximization problem

$$\max_x \sum_{s=1}^{S} \pi_s f_s(r_s^T x) \quad \text{s.t.} \quad Ax \le b \ , \tag{3.1}$$

where f_s is piecewise differentiable (with respect to breakpoint $\hat{y}_s$) for all $s = 1, \dots, S$, by lifting the problem to higher dimension, resulting in a convex program with linear constraints. This chapter follows the treatment in [4].

Defining $B \in \mathbb{R}^{S \times n}$ by $B^T := [r_1, \dots, r_S]$ and introducing new variable $y = Bx \in \mathbb{R}^S$, (3.1) is equivalent to the following problem:

$$\begin{aligned} \max_{x, y=(y_1,\dots,y_S)} \quad & \sum_{s=1}^{S} \pi_s f_s(y_s) \\ & = \sum_{s=1}^{S} \pi_s f_s^{+}\big([y_s - \hat{y}_s]^{+}\big) - \sum_{s=1}^{S} \pi_s f_s^{-}\big([y_s - \hat{y}_s]^{-}\big) + \sum_{s=1}^{S} \pi_s c_s \\ & \text{s.t. } Ax \le b \, , \ y - Bx = 0 \end{aligned} \tag{3.2}$$

Note that the objective function is in general only piecewise differentiable, and the reference points $\hat{y}_s$ are where the objective function is potentially non-differentiable; one way to preclude non-differentiability from the objective function is to split each variable y_s into "positive" and "negative" parts with respect to the reference point $\hat{y}_s$, so that the following relaxed problem may be obtained:

$$\begin{aligned} \max_{x,y,y^+,y^-} \quad & \sum_{s=1}^{S} \pi_s f_s^+(y_s^+) - \sum_{s=1}^{S} \pi_s f_s^-(y_s^-) \\ \text{s.t.} \quad & y^+ - y^- = y - \hat{y} \\ & Ax \leq b\,,\ y - Bx = 0 \\ & y^+, y^- \geq 0 \quad . \end{aligned} \tag{3.3}$$

The variable y in (3.3) can be eliminated, so the problem can be rewritten as

$$\begin{aligned} \max_{x,y^+,y^-} \quad & \sum_{s=1}^{S} \pi_s f_s^+(y_s^+) - \sum_{s=1}^{S} \pi_s f_s^-(y_s^-) \\ \text{s.t.} \quad & y^+ - y^- = Bx - \hat{y}\ ,\ Ax \leq b \\ & y^+, y^- \geq 0\ . \end{aligned} \tag{3.4}$$

In general, the above problem is not equivalent to (3.2). In the special case when f_s^- is a linear function for $s = 1, \ldots, S$, the relaxed problem is equivalent to the original one:

Proposition 3.1 *Suppose f_s^- is linear for $s = 1, \ldots, S$. It follows that (3.2) and (3.3) are equivalent: if (x, y, y^+, y^-) is feasible for (3.3)*[1]*, then $(x, y, [y - \hat{y}]^+, [y - \hat{y}]^-)$ is also feasible for (3.3) and attains an objective value no less than that attained by (x, y, y^+, y^-).*

Proof Suppose (x, y, y^+, y^-) is feasible for (3.3). Then again $(x, y, [y - \hat{y}]^+, [y - \hat{y}]^-)$ is also feasible for (3.3). Moreover,

[1]Henceforth, by abuse of notation, if $a = (a_1, \ldots, a_m)^T \in \mathbb{R}^m$ and $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$, then (a, b) shall be understood as the column vector $(a_1, \ldots, a_m, b_1, \ldots, b_n)^T$.

$$
\begin{aligned}
&\sum_{s=1}^{S} \pi_s \left[f_s^+(y_s^+) - f_s^-(y_s^-) \right] - \sum_{s=1}^{S} \pi_s \left[f_s^+([y_s - \hat{y}_s]^+) - f_s^-([y_s - \hat{y}_s]^-) \right] \\
=& \sum_{s=1}^{S} \pi_s \left[f_s^+(y_s^+) - f_s^+([y_s - \hat{y}_s]^+) \right] - \sum_{s=1}^{S} \pi_s \left[f_s^-(y_s^-) - f_s^-([y_s - \hat{y}_s]^-) \right] \\
\le& \sum_{s=1}^{S} \pi_s f_s^+(y_s^+ - [y_s - \hat{y}_s]^+) - \sum_{s=1}^{S} \pi_s f_s^-(y_s^- - [y_s - \hat{y}_s]^-) \\
=& \sum_{s=1}^{S} \pi_s f_s^+(y_s^+ - [y_s - \hat{y}_s]^+) - \sum_{s=1}^{S} \pi_s f_s^-(y_s^+ - [y_s - \hat{y}_s]^+)
\end{aligned}
$$

which is non-positive by property (P3). The first inequality follows from linearity of f_s^- and concavity of f_s^+: if $y_s^+ > [y_s - \hat{y}_s]^+$, then

$$
\frac{f_s^+(y_s^+) - f_s^+([y_s - \hat{y}_s]^+)}{y_s^+ - [y_s - \hat{y}_s]^+} \le \frac{f_s^+(y_s^+) - f_s^+(0)}{y_s^+ - 0} \le \frac{f_s^+(y_s^+ - [y_s - \hat{y}_s]^+)}{y_s^+ - [y_s - \hat{y}_s]^+}
$$

which implies $f_s^+(y_s^+) - f_s^+([y_s - \hat{y}_s]^+) \le f_s^+(y_s^+ - [y_s - \hat{y}_s]^+)$; else $y_s^+ = [y_s - \hat{y}_s]^+$, which implies $f_s^+(y_s^+) - f_s^+([y_s - \hat{y}_s]^+) = 0 = f_s^+(0) = f_s^+(y_s^+ - [y_s - \hat{y}_s]^+)$. □

Remark. Note that because of property (P3), when f_s^- are all linear, (3.3) has a concave objective function, so (3.2) is trivially equivalent to a convex program.

Even though the lifted problem (3.3) has nice properties (convex and smooth), the problem size grows enormously as the number of scenarios S increases and becomes more and more expensive to solve if general-purpose solvers are directly applied on (3.3). [4] proposed an active set algorithm that, by exploiting duality, circumvents the explicit computations of y^+, y^- except at degenerate iterations, so that the number of scenarios (which is normally much larger than the number of stocks, that is, the length of n) used does not seriously undermine the efficiency of the algorithm.

3.2 Piecewise linear case and duality

This section considers some theoretical aspects in the special case when *both* f_s^+ and f_s^- are linear functions (for $s = 1, \dots, S$). The observations will be the foundation for an economical active set method proposed in [4], which circumvents the explicit computations of y^+, y^-, if the LP at hand is non-degenerate (that is, at every extreme point of dimension $n+2S$, exactly $n+2S$ constraints are active and their gradients are linearly independent). This and the forth-coming sections report some key features of this algorithm.

Suppose that for each $s = 1, \dots, S$,

$$f_s(t) = \begin{cases} p_s \cdot (t - \hat{y}_s) & \text{if } t - \hat{y}_s \geq 0 \\ q_s \cdot (t - \hat{y}_s) & \text{if } t - \hat{y}_s < 0 \end{cases} \quad ; \tag{3.5}$$

where $q_s \geq p_s > 0$ for all s.[2] (It is assumed without loss of generality that $f_s(\hat{y}_s) = 0$ for all s.) Define vectors $\hat{p}, \hat{q} \in \mathbb{R}^S$ by $\hat{p}_s := \pi_s p_s$ and $\hat{q}_s := \pi_s q_s$ for $s = 1, \dots, S$; then (3.4) is equivalent to

$$\begin{aligned} \min_{x, y^+, y^-} \quad & -\hat{p}^T y^+ + \hat{q}^T y^- \\ \text{s.t.} \quad & y^+ - y^- = Bx - \hat{y} \\ & Ax \leq b \\ & y^+, y^- \geq 0 \ ; \end{aligned} \tag{3.6}$$

which is simply a linear program. Its dual is given by

$$\begin{aligned} \max_{\xi, \lambda} \ & \lambda^T \hat{y} - \xi^T b \\ \text{s.t.} \quad & A^T \xi - B^T \lambda = 0 \\ & \hat{p} \leq \lambda \leq \hat{q} \\ & \xi \geq 0 \ , \end{aligned} \tag{3.7}$$

where the free vector λ and non-negative variable ξ in the dual correspond respectively

[2]Here the property (P3) is relaxed slightly: strict inequality $q_s > p_s$ is not enforced here, as the subsequently discussion and algorithms also apply on the more general case when equality is allowed.

to the equality constraint $y^+ - y^- = Bx - \hat{y}$ and the inequality constraint $Ax \leq b$ in the primal. If $R = \{x : Ax \leq b\}$ is nonempty compact, (3.6) has an optimal solution (and so does (3.7)). By complementary slackness, (x, y^+, y^-) is an optimal solution of (3.6) if and only if the Karush-Kuhn-Tucker (KKT) conditions hold:

(1) *Primal feasibility.* $y^+ - y^- = Bx - \hat{y}$, $Ax \leq b$, and y^+, $y^- \geq 0$;

(2) there exist ξ, λ such that

  (a) *Dual feasibility.* $A^T\xi - B^T\lambda = 0$, $\hat{p} \leq \lambda \leq \hat{q}$, and $\xi \geq 0$.

  (b) *Complementary slackness.* $\xi^T(Ax - b) = 0$, $(\lambda - \hat{p})^T y^+ = 0$ and $(\hat{q} - \lambda)^T y^- = 0$.

The KKT conditions can be rephrased in terms of linear systems which will be useful when active set methods are applied on (3.5). For any x feasible for (3.6), define active sets

$$K(x) := \{i : a_i^T x = b_i\}$$
$$J(x) := \{s : r_s^T x = \hat{y}_s\}$$

where a_i denotes the i-th row of A and b_i the i-th entry of b. (Recall $B^T = [r_1, \ldots, r_S]$.)

Observe that a feasible point $(x, [Bx - \hat{y}]^+, [Bx - y]^-)$ is a *non-degenerate* extreme point of the feasible region of (3.6) if and only if $|K(x)| + |J(x)| = n$ and the vectors $\{a_i : i \in K(x)\} \bigcup \{r_s : s \in J(x)\}$ are linearly independent: $(x, y^+, y^-) \in \mathbb{R}^{n+2S}$ is a non-degenerate extreme point if and only if *exactly* $n + 2S$ constraints are active at this point and they have linearly independent gradients. Since there are S equality constraints in the system (and their gradients are linearly independent) and $S + |J(x)|$ sign constraints (for the variables y^+ and y^-) are active at (x, y^+, y^-), $n + 2S$ constraints are active at (x, y^+, y^-) if and only if $n - |J(x)|$ constraints from the inequality system $Ax \leq b$ are active, that is, $|K(x)| = n - |J(x)|$. Since it is assumed that none of the vectors r_s $(s = 1, \ldots, S)$ is zero vector, the active constraints at (x, y^+, y^-) have linearly independent gradients if and only if $\{a_i : i \in K(x)\} \bigcup \{r_s : s \in J(x)\}$ is a linearly independent set of vectors. More generally, since $|K(x)| + |J(x)| + 2S$ constraints are active at a feasible point $(x, [Bx - \hat{y}]^+, [Bx - \hat{y}]^-)$, $(x, [Bx - \hat{y}]^+, [Bx - \hat{y}]^-)$ is an extreme point if and only if $|K(x)| + |J(x)| \geq n$ and $\{a_i : i \in K(x)\} \bigcup \{r_s : s \in J(x)\}$ contains n linearly independent vectors.

Henceforth, (3.6) is *assumed to be non-degenerate* (see Section 3.6 for a further discussion on degeneracy), so that each extreme point $(x, [Bx - \hat{y}]^+, [Bx - \hat{y}]^-)$ of its feasible region is non-degenerate and, consequently, $|K(x)| + |J(x)| = n$.

Let $K(x) =: \{i_1, \ldots, i_{|K(x)|}\}$ and $J(x) =: \{s_1, \ldots, s_{|J(x)|}\}$; define $H(x)$ to be the matrix of active constraint gradients at x:

$$H(x)^T = [a_{i_1}^T, \ldots, a_{i_{|K(x)|}}^T, r_{s_1}^T, \ldots, r_{s_{|J(x)|}}^T]$$

Note that if $(x, [Bx - \hat{y}]^+, [Bx - \hat{y}]^-)$ is a (non-degenerate) extreme point, then $H(x)$ is an invertible matrix by the above observation. Since $i_k \in K(x)$, $a_{i_k}^T x = b_{i_k}$ for each $k = 1, \ldots, |K(x)|$; similarly, $r_{s_j}^T x = \hat{y}_{s_j}$ for each $j = 1, \ldots, |J(x)|$. Hence,

$$H(x) \cdot x = [b_{i_1}, \ldots, b_{i_{|K(x)|}}, \hat{y}_{s_1}, \ldots, \hat{y}_{s_{|J(x)|}}]^T$$

The following result follows directly from the KKT conditions:

Proposition 3.2 *For any $\bar{x} \in \mathbb{R}^n$ satisfying $A\bar{x} \leq b$, define $H = H(\bar{x})$, $\bar{y} = B\bar{x}$, $\bar{y}^+ = [\bar{y} - \hat{y}]^+$ and $y^- = [\bar{y} - \hat{y}]^-$; then $(\bar{x}, \bar{y}^+, \bar{y}^-)$ is optimal for (3.6) if and only if the linear system*

$$H(\bar{x})^T \begin{pmatrix} \xi_{i_1} \\ \vdots \\ \xi_{i_{|K(\bar{x})|}} \\ -\lambda_{s_1} \\ \vdots \\ -\lambda_{s_{|J(\bar{x})|}} \end{pmatrix} = \sum_{s:\, \bar{y}_s > \hat{y}_s} \pi_s p_s r_s + \sum_{s:\, \bar{y}_s < \hat{y}_s} \pi_s q_s r_s \tag{3.8}$$

has a solution $(\bar{\xi}_{i_1}, \ldots, \bar{\xi}_{i_{|K(\bar{x})|}}, -\bar{\lambda}_{s_1}, \ldots, -\bar{\lambda}_{s_{|J(\bar{x})|}})^T$ such that

$$\bar{\xi}_{i_k} \geq 0,\ k = 1, \ldots, |K(\bar{x})| \quad \text{and} \quad \pi_{s_j} p_{s_j} \leq \bar{\lambda}_{s_j} \leq \pi_{s_j} q_{s_j}\,, j = 1, \ldots, |J(\bar{x})|\ . \tag{3.9}$$

Proof If $(\bar{x}, \bar{y}^+, \bar{y}^-)$ is optimal for (3.6), then by the dual feasibility part of the KKT theorem, there exist $\bar{\xi}$, $\bar{\lambda}$ such that $\pi_s p_s \le \bar{\lambda}_s \le \pi_s q_s$ for each $s = 1, \ldots, S$, $\bar{\xi} \ge 0$, and

$$\sum_{i=1}^{m} \bar{\xi}_i a_i - \sum_{s=1}^{S} \bar{\lambda}_s r_s = 0 \ .$$

By the complementary slackness part of the KKT theorem, $\bar{\xi}_i = 0$ for all $i \notin K(\bar{x})$, and $s \notin J(\bar{x})$ implies either $\bar{y}_s > \hat{y}_s$ or $\bar{y}_s < \hat{y}_s$. $\bar{y}_s > \hat{y}_s$ implies $\bar{\lambda}_s = \pi_s p_s$, while $\bar{y}_s < \hat{y}_s$ implies $\bar{\lambda}_s = \pi_s q_s$. By rearranging the terms, (3.8) follows.

For the converse, it suffices to verify that the KKT conditions hold. Primal feasibility follows by definition, so it remains to define $\bar{\xi}$ and $\bar{\lambda}$ that satisfy dual feasibility and complementary slackness. In fact, given the solution $(\bar{\xi}_{i_1}, \ldots, \bar{\xi}_{i_{|K(\bar{x})|}}, -\bar{\lambda}_{s_1}, \ldots, -\bar{\lambda}_{s_{|J(\bar{x})|}})^T$, obtain $\bar{\xi}$ by setting the remaining components to zero:

$$\bar{\xi}_i := 0 \quad \text{for} \quad i \notin K(\bar{x}) \ ;$$

as for $\bar{\lambda}$, since for each $s = 1, \ldots, S$, exactly one of the conditions (1) $s \in J(\bar{x})$; (2) $\bar{y}_s^+ > 0$; or (3) $\bar{y}_s^- > 0$ holds (by definitions of $\bar{y}^+$ and $\bar{y}^-$), obtain $\bar{\lambda}$ from $-\bar{\lambda}_{s_1}, \ldots, -\bar{\lambda}_{s_{|J(\bar{x})|}}$ by setting

$$\bar{\lambda}_j = \begin{cases} \pi_s p_s & \text{if} \quad \bar{y}_s^+ > 0 \\ \pi_s q_s & \text{if} \quad \bar{y}_s^- > 0 \end{cases} \ .$$

Then the complementary slackness conditions and the inequalities in the dual feasibility condition are immediately satisfied; the equality $A^T \bar{\mu} - B^T \bar{\lambda} = 0$ also follows directly from (3.8). Since the KKT conditions are sufficient for optimality of (3.6), $(\bar{x}, \bar{y}^+, \bar{y}^-)$ is an optimal solution of (3.6). □

In particular, if $(\bar{x}, \bar{y}^+, \bar{y}^-)$ is a non-degenerate feasible solution of (3.6), that is, at this point exactly $n + 2S$ constraints with linearly independent gradients are active, then $K(\bar{x}) + J(\bar{x}) = n$ and $H(\bar{x})$ is a $n \times n$ invertible matrix. Based on this observation, it was proposed in [4] that one can iteratively compute the solutions of the system given by (3.8) and check if any of the bound constraints (3.9) on the dual variables is violated: if all are satisfied, an optimal solution is found, else a line search is performed to obtain a new feasible solution.

3.3 Active set algorithm

For the use of the following active set algorithm, assume that the feasible region $R = \{x \in \mathbb{R}^n : Ax \leq b\}$ for (2.6) is compact.[3] It is also assumed that (3.6) is non-degenerate.

Phase 1: Computation of an Initial Feasible Point for (3.6)

Observe that an extreme point x of R corresponds to an extreme point in the feasible region of (3.6). From the linear inequality system defining R alone there are n active constraints with linearly independent gradients; consequently, $|K(x)| + |J(x)| \geq n$ and from the discussion in the last section, the feasible point $(x, [Bx - \hat{y}]^+, [Bx - \hat{y}]^-)$ for (3.6), in particular, has at least $n + 2S$ constraints being active at it, and exactly $n + 2S$ of them are linearly independent. (This can be explicitly given by the n inequalities from $Ax \leq b$, the S equality constraints $y^+ - y^- = Bx - \hat{y}$ and the active sign constraints from $y^+ \geq 0$, $y^- \geq 0$. Note that under the assumption that (3.6) is non-degenerate, for such an extreme point x of R, $J(x)$ must be an empty set, so exactly S of the sign constraints from $y^+ \geq 0$, $y^- \geq 0$ are active.)

This gives a clue for constructing an initial point for solving (3.6). Consider the auxiliary problem

$$\min_{(x,\alpha) \in \mathbb{R}^{n+1}} \alpha \quad \text{s.t.} \quad Ax - b \leq \alpha \mathbb{1} \ , \ \alpha \geq 0 \quad , \tag{3.10}$$

which has an initial feasible point $(0, \max\{|b_i| : i = 1, \ldots, m\})$ (so (3.10) is always feasible). Since R is assumed to be non-empty, applying simplex method on this problem, for instance, returns an optimal solution $(x^0, 0)$[4] and an optimal basis. It is known [5] that

[3] As mentioned in Section 2.1, this is a reasonable assumption since in practice there are bound constraints on the holdings of each asset (such as no-short-sell constraints), and together with the budget constraint the resultant feasible region is indeed a polytope. In particular, R possesses extreme points.

[4] As can be seen here, the prior knowledge that R is non-empty is not really necessary: the auxiliary problem *always* has an optimal solution; the optimal value is 0 if and only if R is nonempty, since any point x in R corresponds to a feasible point $(x, 0)$ for (3.10) which is optimal (by the constraint $\alpha \geq 0$).

this information can be used to generate an extreme point x^0 of R and the corresponding optimal basis K^0, which is of cardinality n. Then $(x^0, [Bx^0 - \hat{y}]^+, [Bx^0 - \hat{y}]^-)$ can be taken as an initial point; take $J^0 = \emptyset$.

Phase 2: Solving (3.6)

Given an extreme point $(x^0, (y^+)^0, (y^-)^0)$, suppose index sets $K^0 := K(x^0)$ and $J^0 := J(x^0)$ of linearly independent active gradients such that $|K^0| + |J^0| = n$ are known. List the elements $K^0 =: \{\alpha_1^0, \ldots, \alpha_{|K^0|}^0\}$, $J^0 =: \{\alpha_{(|K^0|+1)}^0, \ldots, \alpha_n^0\}$ and replace J^0 by $\{\alpha_i^0 + m : \alpha_i^0 \in J^0\}$.

In the algorithm proposed in [4] it is necessary to solve the system (3.8) for the computation of dual variables at each iteration. To improve efficiency, the algorithm works directly with the inverse $H(x^j)^{-1}$. $H(x^j)$ only needs to undergo a change in one row at each iteration, the updating formula Ψ of $H(x^{j+1})^{-1}$ in terms of, $H(x^j)^{-1}$, index k of the row of $H(x^j)$ that is replaced and the new row is explicitly known; see Appendix A.

The active set algorithm proposed in [4] is as follows:

Algorithm 1 Active Set Algorithm for the case when all extreme points are non-degenerate[4]

$j \leftarrow 0$;
$I^0 \leftarrow \{\alpha_1^0, \ldots, \alpha_n^0\}$
$H^0 \leftarrow [h_1, \ldots, h_n]^T$ where the i-th row h_i^T of H^0 is defined by

$$h_i = \begin{cases} a_{\alpha_i^0} & \text{if} \quad 1 \leq \alpha_i^0 \leq m \\ r_{\alpha_i^0 - m} & \text{if} \quad m+1 \leq \alpha_i^0 \leq m+S \end{cases}$$

Algorithm 1 Active set algorithm (continued)

Step 1: Computation of search direction d_j:

Compute vector $t^j := \sum_{s\,:\,r_s^T x^j > \hat{y}_s} \pi_s p_s r_s + \sum_{s\,:\,r_s^T x^j < \hat{y}_s} \pi_s q_s r_s.$

Suppose $(H^j)^{-1} = [c_1^j, \ldots, c_n^j]$; compute indices k_1, k_2, k_3 such that

$$\theta_1^j = (t^j)^T c_{k_1}^j = \min_i \left\{ (t^j)^T c_i^j \,:\, 1 \le \alpha_i^j \le m \right\}$$
$$\theta_2^j = (t^j)^T c_{k_2}^j + \pi_{\alpha_{k_2}^j - m} q_{\alpha_{k_2}^j - m} = \min_i \left\{ (t^j)^T c_i^j + \pi_{\alpha_i^j - m} q_{\alpha_i^j - m} \,:\, m+1 \le \alpha_i^j \le m+S \right\}$$
$$\theta_3^j = -(t^j)^T c_{k_3}^j - \pi_{\alpha_{k_3}^j - m} p_{\alpha_{k_3}^j - m} = \min_i \left\{ -(t^j)^T c_i^j - \pi_{\alpha_i^j - m} p_{\alpha_i^j - m} \,:\, m+1 \le \alpha_i^j \le m+S \right\}$$

Compute $\theta^j = \min\{\theta_1^j, \theta_2^j, \theta_3^j\}$
if $\theta^j \ge 0$ **then**
  STOP; optimal solution found
else
  $k \leftarrow k_l$ where $l \in \{1,2,3\}$ is such that $\theta^j = \theta_l^j$
  $$d^j \leftarrow \begin{cases} c_k^j & \text{if } k = k_1, k_2 \\ -c_k^j & \text{if } k = k_3 \end{cases}$$
end if

Step 2: Computation of step size σ_j.
Compute indices $\tilde{i}$, $\tilde{s}$ such that

$$\sigma_1^j := \frac{a_{\tilde{i}}^T x^j - b_{\tilde{i}}}{a_{\tilde{i}}^T d^j} = \min_i \left\{ \frac{a_i^T x^j - b_i}{a_i^T d^j} \,:\, a_i^T d^j < 0 \right\}$$
$$\sigma_2^j := \frac{r_{\tilde{s}}^T x^j - \hat{y}_{\tilde{s}}}{r_{\tilde{s}}^T d^j} = \min_s \left\{ \frac{r_s^T x^j - \hat{y}_s}{r_s^T d^j} \,:\, \left(r_s^T d^j < 0 \text{ and } r_s^T x^j - \hat{y}_s \le 0\right) \text{ or } \left(r_s^T d^j > 0 \text{ and } r_s^T x^j - \hat{y}_s \ge 0\right) \right\}$$

Compute $\sigma^j = \min\{\sigma_1^j, \sigma_2^j\}$.

Step 3: Update.
$x_{j+1} \leftarrow x_j - \sigma_j d_j$
if $\sigma^j = \sigma_1^j$ **then**
  $(H^{j+1})^{-1} \leftarrow \Psi\left((H^j)^{-1}, a_{\tilde{i}}, k\right)$
  $\alpha_k^{j+1} \leftarrow \tilde{i}$
end if
if $\sigma^j = \sigma_2^j$ **then**
  $(H^{j+1})^{-1} \leftarrow \Psi\left((H^j)^{-1}, r_{\tilde{s}}, k\right)$
  $\alpha_k^{j+1} \leftarrow \tilde{s} + m$
end if
$\alpha_i^{j+1} \leftarrow \alpha_i^j$ for $i \in \{1, \ldots, , k-1, k+1, \ldots, n\}$
$I^{j+1} \leftarrow \{\alpha_1^{j+1}, \ldots, \alpha_n^{j+1}\}$
$j \leftarrow j + 1$
GO TO Step 1

This active set algorithm aims at obtaining a dual feasible solution $(\bar{\xi}, \bar{\lambda})$: recall the dual feasibility condition expressed in Proposition 3.2 that (3.8) given by

$$H(\bar{x})^T \begin{pmatrix} \mu_{i_1} \\ \vdots \\ \mu_{i_{|K(\bar{x})|}} \\ -\lambda_{s_1} \\ \vdots \\ -\lambda y_{s_{|J(\bar{x})|}} \end{pmatrix} = \sum_{s:\, \bar{y}_s > \hat{y}_s} \pi_s p_s r_s + \sum_{s:\, \bar{y}_s < \hat{y}_s} \pi_s q_s r_s$$

has a solution $(\bar{\mu}_{i_1}, \ldots, \bar{\mu}_{i_{|K(\bar{x})|}}, -\bar{\lambda_{s_1}}, \ldots, -\bar{\lambda}_{s_{|J(\bar{x})|}})$ such that (3.9) holds:

$$\bar{\mu}_{i_k} \geq 0,\; k = 1, \ldots, |K(\bar{x})| \quad \text{and} \quad \pi_{s_j} p_{s_j} \leq \bar{\lambda}_{s_j} \leq \pi_{s_j} q_{s_j}\,, j = 1, \ldots, |J(\bar{x})|\;.$$

Based on this result, at each iteration j with an extreme point solution $(x^j, [Bx^j - \hat{y}]^+, [Bx^j - \hat{y}]^-)$ (the fact that this is indeed an extreme point will be proved in the next section) and an invertible matrix $H^j = H(x^j)$ of active constraint gradients at hand, the right hand side of (3.8) is computed and stored in t^j. Then the non-trivial components of dual variables (μ^j, λ^j), which is given by $(H^j)^{-1}t^j$, are computed. If these components lie in the right range (in the sense of (3.9)), then an optimal solution is found; else a search direction is generated from some violated bound constraint on the dual variables: the search direction is taken as a ± 1 multiple of the column of $(H^j)^{-1}$ corresponding to the minimizing index k. It will be shown in the next section that, provided the current iterate is non-degenerate, this is indeed a descent direction. Then a line search is performed so that the new iterate stays feasible, and the constraint gradient matrix H^j (or rather its inverse) as well as the corresponding index set I^j are updated too.

3.4 Key aspects about the algorithm

A few aspects have to be considered in order to claim that the algorithm makes sense. First of all, it has to be shown that each iteration generates an extreme point so that the inverse of the active constraint gradient matrix does exist (and can be used for the next iteration). In addition, it is necessary to show that there is a strict decrease in objective value at each non-degenerate iteration.

Correspondence Between Iterates and Extreme Points

Since the initial point satisfies $Ax^0 \le b$, by the choice of $\sigma^j \le \sigma_1^j$, the algorithm generates a sequence $\{x^j\}_{j=0,1,\ldots}$ of points in R. In fact, each iterate x^j generated by the algorithm corresponds to an extreme point $Y^j = (x^j, [Bx^j - \hat{y}]^+, [Bx^j - \hat{y}]^-)$ of the feasible region of (3.6), provided that the initial point is an extreme point, in order that H^j is defined:

Proposition 3.3 *If the initial iterate* $Y^0 = (x^0, [Bx^0 - \hat{y}]^+, [Bx^0 - \hat{y}]^-)$ *is an extreme point, then each* $Y^j = (x^j, [Bx^j - \hat{y}]^+, [Bx^j - \hat{y}]^-)$ *corresponding to the iterates* x^j *generated by Algorithm 1 is an extreme point of the feasible region of (3.6).*

Proof By construction, Y^0 is an extreme point and H^0 is an invertible n-by-n matrix.

It suffices to show that for each j, the index set I^{j+1} generated at the end of the j-th iteration corresponds to a collection of active constraints at Y^{j+1}; then the corresponding matrix of those active constraint gradients is invertible (see Step 3; this is essentially because the previous matrix is invertible and the new row d^j has a non-zero inner product with the "leaving" column in the inverse of the original matrix) and this means n active constraint gradients at Y^{j+1} are linearly independent.

For a fixed iteration j, suppose k is the leaving index. For $i \ne k$, if $\alpha_i^j \le m$,

$$a_{\alpha_i^j}^T x^{j+1} = b_{\alpha_i^j} - \sigma^j a_{\alpha_i^j}^T d^j \ .$$

Recall that d^j is ± 1 multiple of c_j^k, which is the k-th column of $(H^j)^{-1}$, and $a_{\alpha_i^j}^T$ is the i-th row of H^j, $a_{\alpha_i^j}^T d^j$ must be zero. Therefore $a_{\alpha_i^j}^T x^{j+1} = b_{\alpha_i^j}$.

If $\alpha_i^j > m$,

$$r_{\alpha_i^j - m}^T x^{j+1} = \hat{y}_{\alpha_i^j - m} - \sigma^j r_{\alpha_i^j - m}^T d^j$$

and since $r_{\alpha_i^j - m}^T$ is the i-th row of H^j, for the same reason $r_{\alpha_i^j - m}^T d^j = 0$, and $r_{\alpha_i^j - m}^T x^{j+1} =$

$\hat{y}_{\alpha_i^j - m}$.

Finally, if α_k^{j+1} is taken to be $\tilde{i}$ in Step 3, then σ^j must equal $\dfrac{a_{\tilde{i}}^T x^j - b_{\tilde{i}}}{a_{\tilde{i}}^T d^j}$, which means $b_{\tilde{i}} = a_{\tilde{i}}^T x^{j+1}$. If α_k^{j+1} is taken to be $\tilde{s}$ instead, then σ^j must equal $\dfrac{r_{\tilde{s}}^T x^j - \hat{y}_{\tilde{s}}}{r_{\tilde{s}}^T d^j}$, which means $r_{\tilde{s}}^T x^{j+1} = \hat{y}_{\tilde{s}}$. In either case, the n constraints collected by the matrix H^{j+1} are active at Y^{j+1}. Therefore Y^{j+1} is again an extreme point. □.

Improvement at non-degenerate iterations

The next task is to prove that the search direction generated by Algorithm 1 indeed leads to strict decrease in objective value provided the current iterate is non-degenerate. The following result will be useful for proving the claim:

Lemma 3.4 *Given an iterate x^j, if at state s, $r_s^T x^j < \hat{y}_s$, the next iterate must satisfy $r_s^T x^{j+1} \le \hat{y}_s$. If $r_s^T x^j > \hat{y}_s$, the next iterate must satisfy $r_s^T x^{j+1} \ge \hat{y}_s$.*

Proof Suppose $r_s^T x^j > \hat{y}_s$ and the conclusion does not hold; then $r_s^T x^j > \hat{y}_s > r_s^T x^{j+1}$. Since $x^{j+1} = x^j - \sigma^j d^j$ and the current iteration is non-degenerate, $\sigma^j > 0$ and so $r_s^T d^j > 0$. By definition of σ_2^j,

$$\sigma^j \le \sigma_2^j \le \frac{r_s^T x^j - \hat{y}_s}{r_s^T d^j}$$

By rearranging the terms, it follows that $\hat{y}_s \le r_s^T (x^j - \sigma^j d^j) = r_s^T x^{j+1}$, which is contradictory. Therefore if $r_s^T x^j > \hat{y}_s$ holds, the next iterate must satisfy $r_s^T x^{j+1} \ge \hat{y}_s$.

By the same argument, if $r_s^T x^j < \hat{y}_s$ holds, the next iterate must satisfy $r_s^T x^{j+1} \le \hat{y}_s$. □

Theorem 3.5 *Suppose $Y^j = (x^j, (y^j)^+, (y^j)^-)$, where $(y^j)^+ = [Bx^j - \hat{y}]^+$ and $(y^j)^- = [Bx^j - \hat{y}]^-$ is a non-degenerate extreme point of (3.6). Let x^{j+1} be the new iterate generated from x^j by Algorithm 1 and $Y^{j+1} = (x^{j+1}, (y^{j+1})^+, (y^{j+1})^-)$, where $(y^{j+1})^+ = [Bx^{j+1} - \hat{y}]^+$ and $(y^{j+1})^- = [Bx^{j+1} - \hat{y}]^-$; then*

$$\Delta^j := -\hat{p}^T((y^{j+1})^+ - (y^j)^+) + \hat{q}^T((y^{j+1})^- - (y^j)^-) < 0$$

Proof Since

$$-\hat{p}^T(y^j)^+ + \hat{q}^T(y^j)^- = -\sum_{s\,:\,r_s^T x^j > \hat{y}_s} \pi_s p_s(r_s^T x^j - \hat{y}_s) + \sum_{s\,:\,r_s^T x^j < \hat{y}_s} \pi_s q_s(\hat{y}_s - r_s^T x^j) \quad \text{, and}$$

$$\begin{aligned}
-\hat{p}^T(y^{j+1})^+ + \hat{q}^T(y^{j+1})^- &= \left(\sum_{s\,:\,r_s^T x^j > \hat{y}_s} + \sum_{s\,:\,r_s^T x^j = \hat{y}_s} + \sum_{s\,:\,r_s^T x^j < \hat{y}_s} \right) \left(-\pi_s p_s (y^{j+1})_s^+ + \pi_s q_s (y^{j+1})_s^-\right) \\
&= -\sum_{s\,:\,r_s^T x^j > \hat{y}_s} \pi_s p_s(r_s^T x^{j+1} - \hat{y}) + \sum_{s\,:\,r_s^T x^j < \hat{y}_s} \pi_s q_s(\hat{y} - r_s^T x^{j+1}) \; + \; \delta
\end{aligned}$$

where the last line follows from Lemma (3.4) and

$$\begin{aligned}
\delta :=& -\sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s p_s (y^{j+1})_s^+ + \sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s q_s (y^{j+1})_s^- \\
=& -\sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s p_s [r_s^T x^{j+1} - \hat{y}_s]^+ + \sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s q_s [r_s^T x^{j+1} - \hat{y}_s]^- \\
=& -\sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s p_s [-\sigma^j r_s^T d^j]^+ + \sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s q_s [-\sigma^j r_s^T d^j]^- \\
=& -\sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s p_s \sigma^j [r_s^T d^j]^- + \sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s q_s \sigma^j [r_s^T d^j]^+ \; .
\end{aligned}$$

Note that δ takes three different values depending on the choice of k:

(1) $k = k_1 \in \{1, \ldots, m\}$: then $d^j = c_k^j$ is a column in $(H^j)^T$ corresponding to some constraint from the system $Ax \leq b$, so $r_s^T d^j = 0$ for all s, and $\delta = 0$.

(2) $k = k_2$: since $r_s^T c_k^j = 1$ only if $s = \alpha_k^j - m$ and is zero otherwise when $k = k_2$ or k_3, and in the present case $d^j = c_k^j$, it follows that

$$\begin{aligned}
\delta =& -\sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s p_s \sigma^j [r_s^T c_{k_2}^j]^- + \sum_{s\,:\,r_s^T x^j = \hat{y}_s} \pi_s q_s \sigma^j [r_s^T c_{k_2}^j]^+ \\
=& \pi_{\alpha_{k_2}^j - m} q_{\alpha_{k_2}^j - m} \sigma^j \quad .
\end{aligned}$$

(3) $k = k_3$: since $d^j = -c_k^j$,

$$
\begin{aligned}
\delta =& - \sum_{s\,:\, r_s^T x^j = \hat{y}_s} \pi_s p_s \sigma^j [r_s^T c_{k_3}^j]^+ + \sum_{s\,:\, r_s^T x^j = \hat{y}_s} \pi_s q_s \sigma^j [r_s^T c_{k_3}^j]^- \\
=& - \pi_{\alpha_{k_3}^j - m} p_{\alpha_{k_3}^j - m} \sigma^j \quad .
\end{aligned}
$$

It follows that

$$
\begin{aligned}
\Delta^j =& - \sum_{s\,:\, r_s^T x^j > \hat{y}_s} \pi_s p_s \big((r_s^T x^{j+1} - \hat{y}) - (r_s^T x^j - \hat{y}_s) \big) + \sum_{s\,:\, r_s^T x^j < \hat{y}_s} \pi_s q_s \big((\hat{y} - r_s^T x^{j+1}) - (\hat{y}_s - r_s^T x^j) \big) + \delta \\
=& \sum_{s\,:\, r_s^T x^j > \hat{y}_s} \pi_s p_s \sigma^j r_s^T d^j + \sum_{s\,:\, r_s^T x^j < \hat{y}_s} \pi_s q_s \sigma^j r_s^T d^j + \delta \\
=& \begin{cases} \sigma^j (t^j)^T d^j \quad \text{if} \quad k = k_1 \\ \sigma^j \left[(t^j)^T d^j + \pi_{\alpha_{k_2}^j - m} q_{\alpha_{k_2}^j - m} \right] \quad \text{if} \quad k = k_2 \\ \sigma^j \left[(t^j)^T d^j - \pi_{\alpha_{k_3}^j - m} p_{\alpha_{k_3}^j - m} \right] \quad \text{if} \quad k = k_3 \end{cases} \\
=& \sigma^j \theta^j \quad .
\end{aligned}
$$

By non-degeneracy assumption, it follows that $\sigma^j > 0$. On the other hand, to be able to generate the new iterate x^{j+1}, θ^j must be negative (otherwise the algorithm would have stopped before that). Therefore Δ^j, the change in objective value, is negative. □

As a consequence of this theorem, Algorithm 1 guarantees finite termination under the assumption that (3.6) is non-degenerate.

3.5 Numerical results

The following shows some computation results on some sample problems solved by the active set algorithm proposed in [4]. The algorithm was programmed in Fortran 77 and compiled with the Watcom compiler (version 10). The computer CPU used was an Intel Pentium M Processor 740, running at 1.73 GHz.

Key:

(1) *Assets*: number of assets

(2) *States*: number of states

(3) *Constraints*: number of linear inequality constraints

(4) *First slope*: slope of the first linear piece (corresponding to returns below reference point)

(5) *Second slope*: slope of the second linear piece (all normalized to one)

(6) *Iterations*: number of iterations

Problem	Assets	States	Constraints	First slope	Second slope	Iterations
C02	14	319	30	2.25	1	90
C05	12	78	26	3.5	1	24
C10a	12	78	26	10	1	31
C10b	12	78	26	7.5	1	30
C10c	12	78	26	5.0	1	26
C10d	12	78	26	4.5	1	24
C10e	12	78	26	4.0	1	22
C10f	12	78	26	3.5	1	24
C10g	12	78	26	3.0	1	2
C10h	12	78	26	2.5	1	3
C10i	12	78	26	2.25	1	3
C10j	12	78	26	2.0	1	3
C10k	12	78	26	1.5	1	3
C10l	12	78	26	1.25	1	3

Remark. Problems C10a-l use the same set of linear constraints and state data. The only difference among them is the slope of the first piece.

3.6 What about degeneracy?

Obviously, the assumption that (3.6) is non-degenerate is a rather restrictive one. In fact, in practice the number of states S is enormous, and degeneracy is almost inevitable. So is there a way to handle degeneracy in the framework of Algorithm 1?

One might ask if implementing typical anti-cycling rules, such as Bland's rule (which is also called smallest-subscript rule) could solve the problem. Unfortunately, the duality result suggests that it is not quite possible to directly apply Bland's rule. Recall, from the necessary and sufficient condition of optimality for $(x, [Bx - \hat{y}]^+, [Bx - \hat{y}]^-)$ from Proposition 3.3, that the system

$$\sum_{i \in K(x)} \xi_i a_i - \sum_{s \in J(x)} \lambda_s r_s = \sum_{s:\, \bar{y}_s > \hat{y}_s} \pi_s p_s r_s + \sum_{s:\, \bar{y}_s < \hat{y}_s} \pi_s q_s r_s \quad . \tag{3.11}$$

This equation is crucial for Algorithm 1 to compute the dual variables. Under non-degeneracy assumption, for each extreme point there corresponds a unique dual solution that can be computed via the above equation, and by checking feasibility of the dual solution the algorithm determines if the current iterate is optimal.

Before discussing the difficulty in the presence of degeneracy, note that there is a subtlety in (3.11): at each iteration j of Algorithm 1, there is a working set $I^j = K^j \bigcup (J^j + m)$, where K_j collects the indices i in I^j that correspond to the active constraint $a_i^T x = b_i$ and J^j collects the indices s (in I^j, by abuse of notation) that correspond to the active constraint $r_s^T x = \hat{y}_s$. The algorithm in effect aims at solving (3.11)

$$\sum_{i \in K} \xi_i a_i - \sum_{s \in J} \lambda_s r_s = -\sum_{i \in K(x) \setminus K} \xi_i a_i + \sum_{s \in J(x) \setminus J} \lambda_s r_s + \sum_{s:\, \bar{y}_s > \hat{y}_s} \pi_s p_s r_s + \sum_{s:\, \bar{y}_s < \hat{y}_s} \pi_s q_s r_s \tag{3.12}$$

for ξ_i $(i \in K)$ and λ_s $(s \in J)$, assuming that the right-hand side is known *a priori*. But this is usually not the case when degeneracy occurs.

To see the difficulty that would arise in the presence of degeneracy, consider the case when the algorithm reaches a degenerate optimal solution $(\bar{x}, [B\bar{x}-\hat{y}]^+, [B\bar{x}-\hat{y}]^-)$ with working sets $K \subseteq K(x)$ and $J \subseteq J(x)$. This optimal solution would correspond to some Lagrange multiplier $(\bar{\xi}, \bar{\lambda})$ that satisfies the dual feasibility condition and complementary slackness. The multiplier does not necessarily satisfy $\bar{\xi}_i = 0$ for $i \in K(\bar{x})\backslash K$, and $\bar{\lambda}_s = \pi_s p_s$ or $\pi_s q_s$ for $s \in J(\bar{x})\backslash J$. Even if there are Lagrange multipliers that satisfy this, the algorithm does not seem to be able to produce the right multipliers whose indices are outside the working sets *a priori*. If one tries to impose particular values on ξ_i for $i \in K(\bar{x})\backslash K$ or λ_s for $s \in J(\bar{x})\backslash J$ and solve for (3.12), the solution obtained may not necessarily satisfy the bound constraints. By failing to produce the true certificate of optimality, the algorithm would cycle, *even if anti-cycling rules such as Bland's rule is incorporated in the algorithm*. As such, to guarantee finite termination in the case when (3.6) is degenerate, not only anti-cycling rule, but also a way of determining the right multipliers for those active constraints not indexed by the working set, is necessary, and is one subject for further research.

Chapter 4

Smoothing Technique: an outline

4.1 Smoothing the non-differentiable objective function

An alternative for dealing with the non-smooth problem (3.6) is to solve an approximate problem that has a smooth objective function.

Respectively in the frameworks of CVaR minimization with transaction costs and mean-variance portfolio optimization with piecewise linear transaction costs, [2] and [18] proposed different methods to solve minimization problems with separable piecewise linear convex objective function and linear constraints. The first step taken in both works is to "smooth" out all the "corners" of the graph (due to the existence of break points) by forming a convex quadratic/cubic approximation around the corners, so that the resultant function is smooth, and can the approximate the original objective function as well as one wishes. By solving the approximate problem, one gets an approximate solution to the original problem. Details on the fact that the (approximate) solution obtained from the new smooth problem is indeed a good proxy for the original problem can be found in [9].

To see how the smoothing technique works, take as an illustration the piecewise linear concave utility function from (3.5):

$$f_s(t) = \begin{cases} p_s \cdot (t - \hat{y}_s) & \text{if } t - \hat{y}_s \geq 0 \\ q_s \cdot (t - \hat{y}_s) & \text{if } t - \hat{y}_s < 0 \end{cases}$$
$$= p_s[t - \hat{y}_s]^+ - q_s[t - \hat{y}_s]^- \quad ,$$

where q_s is assumed without loss of generality to be strictly greater than $p_s > 0$ (otherwise f_s is simply a linear function). Fixing a *resolution parameter* $\varepsilon > 0$ that determines how close the approximation is going to be, the continuously differentiable piecewise quadratic approximation $\tilde{f}_s(t; \varepsilon)$ is defined by

$$\tilde{f}_s(t; \varepsilon) := f_s(t) - \sigma_s(t; \varepsilon)$$

where the function $\sigma_s(\cdot; \varepsilon)$ is such that it vanishes outside $[-\varepsilon, \varepsilon]$ and the resultant approximation $\tilde{f}_s(\cdot; \varepsilon)$ is sufficiently smooth. When four parameters ($f_s(\pm\varepsilon)$ and $f_s'(\varepsilon)$) are available, f_s can be approximated either by quadratic or cubic *splines*.

If, for instance, quadratic spline is desired, σ_s can be given by

$$\sigma_s(t; \varepsilon) := \begin{cases} \frac{q_s - p_s}{4\varepsilon}(t - \hat{y}_s + \varepsilon)^2 & \text{if } t - \hat{y}_s \in [-\varepsilon, 0] \\ \frac{q_s - p_s}{4\varepsilon}(t - \hat{y}_s - \varepsilon)^2 & \text{if } t - \hat{y}_s \in [0, \varepsilon] \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

(Alternatively, a convex cubic spline can be given by

$$\sigma_s(t; \varepsilon) := \begin{cases} \frac{q_s - p_s}{6\varepsilon^2}(t - \hat{y}_s + \varepsilon)^3 & \text{if } t - \hat{y}_s \in [-\varepsilon, 0] \\ -\frac{q_s - p_s}{6\varepsilon^2}(t - \hat{y}_s - \varepsilon)^3 + (q_s - p_s)t & \text{if } t - \hat{y}_s \in [0, \varepsilon] \\ 0 & \text{otherwise .} \end{cases} \tag{4.2}$$

)

Then the optimal solution of the problem

$$\begin{aligned} \max_x \quad & \sum_{s=1}^{S} \pi_s f_s(r_s^T x) \\ \text{s.t.} \quad & Ax \leq b \end{aligned} \tag{4.3}$$

can be approximated by solving the smooth piecewise quadratic optimization problem

$$\begin{aligned} \max_{x} \quad & \sum_{s=1}^{S} \pi_s \tilde{f}_s(r_s^T x; \varepsilon) \\ \text{s.t.} \quad & Ax \leq b \quad , \end{aligned} \tag{4.4}$$

which is still defined only on n variables, so again the number of states would not directly affect the efficiency of solving each of these problems. The optimal objective value of (4.4) differs from that of (4.3) only by at most $(q_s - p_s)\varepsilon/4$ (see Appendix B).

Depending on the algorithm used to solve the smooth problem, either quadratic or cubic spline may be used. In [2], a special but mainstream class of minimization problem with piecewise linear convex objective function, linear equality and bound constraints is considered; authors of [2] proposed to use a specialized trust region method to solve the problem, so quadratic spline is naturally preferred as the trust region method builds quadratic approximation of the objective function in the trust region, and by using quadratic spline the trust region subproblem quite often has the original objective function as its own objective. On the other hand, in [18], interior point method is used (in locating the initial point for the subsequent application of active set method); since each search direction is obtained by solving a Newton's equation, quadratic splines does not offer overwhelming advantage over cubic splines, so cubic spline could be used for a better approximation.

4.2 Double trust region method

In the present case when the original non-smooth function is piecewise linear, the resultant approximating problem (4.4) is indeed a smooth convex program problem.

Based on this fact that the resultant approximating problem (4.4) is a smooth convex program problem (by changing the sign of the objective function to get a minimization problem), [2] focused on a special case when the set of constraints consists of (normally very few, probably just one or two) equality constraints and bound constraints, that is, a problem of the form

$$\begin{aligned} \min_{x} \quad & -\sum_{s=1}^{S} \pi_s \tilde{f}_s(r_s^T x; \varepsilon) \\ \text{s.t.} \quad & Ax = b \quad , \quad l \le x \le u \quad , \end{aligned} \tag{4.5}$$

where $A \in \mathbb{R}^m \times n$, $b \in \mathbb{R}^m$, and $l, u \in \mathbb{R}^n$ with $l < u$. To solve this problem, the authors in [2] implemented an interior point method based on a specialized trust region method [6] for nonlinear minimization with bound constraints. It has been shown that, under mild assumptions, this trust region method can achieve global convergence and local quadratic convergence.

The smoothing technique can achieve very impressive speed even on large-scale problems; this is a crucial merit especially because, as a result of the states being quite often generated by Monte Carlo simulation, the problem size is normally very huge (not only in terms of the number of states but also the number of instruments). As an illustration, below is an extract of CPU times results obtained from running Mosek on the lifted version of CVaR minimization problem (2.12) (the lifted version is similar to (3.6)) versus the double trust region method on problem (4.3):

No. of scenarios	Mosek (CPU sec)			Smoothing (CPU sec)		
	No. of instruments being considered					
	8	48	200	8	48	200
10,000	6.47	42.04	4244.30	2.45	16.78	419.52
25,000	33.50	98.91	10,784.10	5.27	35.48	838.15
50,000	36.01	318.72	-	9.90	62.08	2080.16

Source: [2], P. 598

One advantage of the smoothing technique is that it can be easily extended to the case where f_s^- is a linear function, as the resultant utility function is still a separable piecewise concave function.

4.3 Interior point method and cross-over technique

In the last sections, smoothing technique is introduced to obtain a good approximate solution to the original non-smooth problem efficiently. Smoothing technique alone, however, does not give an exact solution, nor does it have finite termination properties like active set methods do. To get the merits from both methods, [18] proposed a cross-over method: in the present case, interior point method can be applied on the smooth problem to approximately solve the original problem, and the approximate solution is then used to locate an exact solution via standard active set method. The idea is that interior point method, which is known to be efficient and capable of dealing with large-scale problems, can help getting a near-optimal point as initial point for running the active set method, so that at the end an exact solution can be obtained without visiting too many extreme points in vain.

Strategy proposed in [18] is designed to deal with a convex program with an objective function that is the sum of a quadratic function and a separable piecewise linear function of x, together with linear constraints. Yet, with the development of interior point method for convex program [19] and more recently, for non-linear program (see [10] and [16], for example), it might be inspiring to phrase the idea in a more general framework. Again recall the non-smooth problem (3.1) of interest, put as a minimization problem

$$\min_x F(x) = -\sum_{s=1}^{S} \pi_s f_s(r_s^T x) \quad \text{s.t.} \quad Ax \leq b \ , \tag{4.6}$$

where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. The Lagrangian of this problem is defined as $L(x,\xi) := F(x) + \xi^T(Ax-b)$ (with $\xi \in \mathbb{R}^m$). The KKT conditions of the problem are given by

Dual feasibility.	$0 \in \partial_x L(x,\xi) \quad , \ \xi \geq 0$
Primal feasibility.	$Ax \leq b$
Complementary slackness.	$\xi^T(Ax-b) = 0$.

By subdifferential calculus rules [21], the condition $0 \in L(x,\xi)$ is equivalent to $A^T\xi \in \sum_{s=1}^{S} \pi_s \partial_x f_s(r_s^T x)$, where

$$\partial_x f_s(r_s^T x) = \begin{cases} \left\{ \frac{d}{dt} f^+([r_s^T x - \hat{y}_s]^+) r_s \right\} & \text{if } r_s^T x - \hat{y}_s > 0 \\ \left\{ -\frac{d}{dt} f^-([r_s^T x - \hat{y}_s]^-) r_s \right\} & \text{if } r_s^T x - \hat{y}_s < 0 \\ \left\{ \gamma r_s : \gamma \in \left[-\frac{d}{dt} f^-(0^-), \frac{d}{dt} f^+(0^+) \right] \right\} & \text{if } r_s^T x - \hat{y}_s = 0 \ . \end{cases}$$

Here $\frac{d}{dt} f^-(0^-)$ and $\frac{d}{dt} f^+(0^+)$ denote the left-hand derivative of f^- and right-hand derivative of f^+ at zero respectively. If F is convex, the KKT conditions are sufficient for local optimality too.

If each of the non-smooth functions f_s is to be approximated by $\tilde{f}_s(\cdot;\varepsilon)$, the new approximate problem

$$\min_x \tilde{F}(x) := -\sum_{s=1}^{S} \pi_s \tilde{f}_s(r_s^T x; \varepsilon) \quad \text{s.t.} \quad Ax \leq b \ , \tag{4.7}$$

is smooth and the KKT conditions are quite similar, except that the complication (that is, the subdifferential) disappears:

Dual feasibility. $\quad A^T \xi = \sum_{s=1}^{S} \pi_s \tilde{f}_s'(r_s^T x) r_s \quad , \ \xi \geq 0$

Primal feasibility. $\quad Ax \leq b$

Complementary slackness. $\quad \xi^T (Ax - b) = 0$

A typical interior point method for solving the smooth problem seeks to solve the perturbed KKT system

$$\begin{aligned} &\sum_{s=1}^{S} \pi_s \tilde{f}_s'(r_s^T x) r_s - A^T \xi = 0 \\ &Ax + \eta = b \\ &\xi_i \eta_i = \mu \\ &\xi > 0 \ , \ s > 0 \ . \end{aligned}$$

Here $\mu := \xi^T \eta / n$ is the barrier parameter[1]. An interior point method finds a search

[1] Unfortunately, the popular Greek letter μ is commonly used in finance to refer to expected return

direction from the current iterate by solving the linearized KKT system (around the current iterate) and perform an appropriate line search to maintain strict feasibility. The KKT system linearized at (x, ξ, η) is given by

$$\begin{pmatrix} G & A^T & 0 \\ A & 0 & I \\ 0 & \mathrm{Diag}(\eta) & \mathrm{Diag}(\xi) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \xi \\ \Delta \eta \end{pmatrix} = \begin{pmatrix} -\rho_d \\ -\rho_p \\ -\xi \circ \eta + \sigma \mu \mathbb{1} \end{pmatrix}$$

where ρ_d and ρ_p are the residuals

$$\rho_d = \sum_{s=1}^{S} \pi_s \tilde{f}_s'(r_s^T x) r_s - A^T \xi \quad , \quad \rho_p = Ax + \eta - b \ ,$$

$G := \sum_{s=1}^{S} \pi_s \tilde{f}_s''(r_s^T x) r_s r_s^T$ is the Hessian of the objective function, $\xi \circ \eta$ denotes the Hadamard product (or component-wise product) of the vectors ξ and η, and $\sigma \in [0, 1]$ is the centering parameter.

In general, by choosing an initial point, and then iteratively computing a search direction and appropriate step size for updating the current iterate, an approximate solution to (4.7) (for some fixed $\varepsilon > 0$)) can be computed efficiently. [18] proposed that *purification step* be then performed, that is, several iterations of gradient projection method be applied starting from the solution from interior point method in order to increase the size of active set without increasing the objective value. Finally, a standard active set method [9] is applied starting from the last iterate obtained from the purification step.

In this cross-over strategy, the application of interior point method does not aim at a very precise solution; since the last iterate is used only as an initial point for purification, both the resolution parameter ε and the tolerance used in the interior point method can be crude. This in principle could reduce the number of iterations of interior point method, which is very advantageous as every step is quite expensive to compute. The number of iterations in the stage of active set method is also much smaller than what one would

vector of assets and in numerical optimization as barrier parameter. Since in this section all focus is solely on interior point methods, μ is to be understood as barrier parameter all through this section and hopefully no confusion would arise.

expect from a straight forward application of standard active set method.

[18] proposed the above cross-over technique in the setting of convex program, and the Hessian G in their problem is the sum of a constant matrix and a non-constant diagonal matrix. Even though the Hessian is not constant overall, the computational effort is not as great as in general case because of the choice of simple splines. It is not known if cross-over technique would perform equally well on the present problem (4.6), even when it is a convex program, let alone one with non-convex objective function due to the S shape of utility functions. Greater care is also needed in such case in computing the Hessian and the step size. Yet the cross-over technique, by taking credit from both interior point methods and active set methods, seems to be a good option for solving a non-smooth and yet fairly well-structured problem like (4.6).

Bibliography

[1] T. Adler and M. Kritzman. Mean-variance versus full-scale optimization: in and out of sample. *Journal of Asset Management*, 7:302–311, 2007. 5

[2] S. Alexander, T.F. Coleman, and Y. Li. Minimizing cvar and var for a portfolio of derivatives. *Journal of Banking and Finance*, 30(2):583–605, 2006. 31, 33, 34

[3] A. Berkelaar and R. Kouwenberg. Optimal portfolio choice under loss aversion. *Review of Economics and Statistics*, 84:973–987, 2000. 5

[4] M. J. Best, R. Grauer, and J. Hlouskova. Loss-aversion with kinked linear utility functions, 2009. 13, 15, 16, 19, 21, 28

[5] M. J. Best and K. Ritter. *Linear programming: active set analysis and computer programs*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1985. 20, 41

[6] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996. 34

[7] J. Cremers, M. Kritzman, and S. Page. Optimal hedge fund allocations. *Journal of Portfolio Management*, 31(3):70–81, 2005. 5

[8] J. Cremers, M. Kritzman, and S. Page. Portfolio formation with higher moments and plausible utility. *Revere Street Working Paper Series*, (272-12), 2007. 5

[9] A. V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, New York :, 1983. 31, 37

[10] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. on Optimization*, 8(4):1132–1152, 1998. 35

[11] B. Hagstromer, R. Anderson, J. Binner, T. Elger, and B. Bilsson. Mean-variance versus full-scale optimization: broad evidence for the UK. *FRB of St. Louis Working Paper*, (2007-016D), 2008. 5

[12] D. Kahneman and A. Tversky. Prospect theory: an analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979. 2, 6, 8

[13] H. Levy and H. M. Markowitz. Approximating expected utility by a function of mean and variance. *The American Economic Review*, 69(3):308–317, 1979. 4

[14] H. Markowitz. *Mean-variance analysis in portfolio choice and capital markets*. Wiley, John and Sons, Incorporated, Blackwell, 1987. 4

[15] H. Markowitz. Portfolio selection. *Journal of Finance*, VII(1):77–91, March 1952. 1

[16] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, August 2006. 35

[17] G. Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. In S. Uryasev, editor, *Probabilistic constrained optimization*. Kluwer Academic Publishers, 2000. 10

[18] M. Potaptchik, L. Tunçel, and H. Wolkowicz. Large scale portfolio optimization with piecewise linear transaction costs. *Optimization Methods and Software*, 23(6):929–952, 2008. 31, 33, 35, 37, 38

[19] F. A. Potra and S. J. Wright. *Interior-Point Methods*. SIAM, 2000. 35

[20] R. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–42, 2000. 2, 10, 11

[21] R. T. Rockafellar. *Convex Analysis (Princeton Mathematical Series)*. Princeton University Press, 1996. 35

[22] F. Sortino and S. Satchell, editors. *Managing downside risk in financial markets*. Butterworth-Heinemann, 1st edition, 2001. 5

Appendix A

Updating formula for matrix inverse

This appendix gives the explicit updating formula Ψ of an inverse of a matrix when one of its rows is replaced by some other vector, following the treatment in [5]. Ψ requires $O(n^2)$ arithmetic operations.

Let $X^T = [u_1, \ldots, u_n]$ be an $n \times n$ nonsingular matrix and suppose $X^{-1} = [v_1, \ldots, v_n]$ (so that for any $i, j \in \{1, \ldots, n\}$, $u_i^T v_j = \delta_{ij}$ where δ_{ij} equals one when $i = j$ and is zero otherwise).

Suppose u is an n-vector that is to replace the k-th column of X^T. Provided $u^T v_k \neq 0$, the new matrix is invertible. To start with, denote the new matrix by $\hat{X}$:

$$\hat{X}^T := [\hat{u}_1, \ldots, \hat{u}_n] := [u_1, \ldots, u_{k-1}, u, u_{k+1}, \ldots, u_n] \ .$$

Then the matrix $[\hat{v}_1, \ldots, \hat{v}_n] \in \mathbb{R}^{n \times n}$, where each column $\hat{v}_j$ is defined by

$$\hat{v}_j = \begin{cases} \frac{1}{u^T v_k} v_k & \text{if} \quad j = k \\ v_j - \frac{u^T v_j}{u^T v_k} v_k & \text{otherwise} \end{cases} ,$$

is the inverse of $\hat{X}$. To prove this claim, it is necessary to show that $\hat{u}_i^T \hat{v}_j = \delta_{ij}$. For $i \neq k$,

$$\hat{u}_i^T \hat{v}_j = \begin{cases} u_i^T(\frac{1}{u^T v_k} v_k) & \text{if} \quad j = k \\ u_i^T(v_j - \frac{u^T v_j}{u^T v_k} v_k) & \text{otherwise} \end{cases}$$

$$= \begin{cases} 0 & \text{if} \quad j = k \\ u_i^T v_j - \frac{u^T v_j}{u^T v_k} u_i^T v_k & \text{otherwise} \end{cases}$$

$$= \begin{cases} 0 & \text{if} \quad j \neq i \\ u_j^T v_j - \frac{u^T v_j}{u^T v_k} u_j^T v_k = 1 & \text{otherwise} \end{cases}$$

$$= \delta_{ij} \ .$$

For the case when $i = k$,

$$\hat{u}_k^T \hat{v}_j = \begin{cases} u^T(\frac{1}{u^T v_k} v_k) & \text{if} \quad j = k \\ u^T v_j - \frac{u^T v_j}{u^T v_k} u^T v_k & \text{otherwise} \end{cases}$$

$$= \begin{cases} 1 & \text{if} \quad j = k \\ 0 & \text{otherwise} \end{cases}$$

$$= \delta_{kj} \ .$$

Hence $\hat{u}_i^T \hat{v} = \delta_{ij}$, that is, $(\hat{X})^{-1} = [\hat{v}_1, \ldots, \hat{v}_n]$.

In conclusion, it is possible to define an inverse operator Ψ for any tuple (X, u, k), where $X = [u_1, \ldots, u_n]^T$ is an $n \times n$ invertible matrix with $X^{-1} = [v_1, \ldots, v_n]$, $k \in \{1, \ldots, n\}$ and $u \in \mathbb{R}^n$ with the property that its inner product with the k-th row of X is non-zero: $\Psi(X, u, k)$ is the inverse of the new matrix obtained from X by replacing its k-th row by vector u, and the formula is given by

$$\Psi(X, u, k) = [\hat{v}_1, \ldots, \hat{v}_n] \quad \text{where} \quad \hat{v}_i = \begin{cases} \frac{1}{u^T v_k} v_k & \text{if} \quad i = k \\ v_i - \frac{u^T v_i}{u^T v_k} v_k & \text{otherwise} \end{cases} .$$

Remark. In fact, the condition $u^T v_k \neq 0$ is necessary for the new matrix to be invertible: if $u^T v_k = 0$, then v_k is in the null space of the new matrix; since v_k is a column of an invertible matrix, v_k is non-zero. This shows that the new matrix has a non-trivial null space, and is therefore singular.

Appendix B

Remarks on the approximation of piecewise linear function

This appendix gives a simple error bound of the quadratic approximation $\tilde{f}_s(t;\varepsilon)$ of $f_s(t)$ given in Section 4.1.

The quadratic spline given in Section 4.1 could be derived from the following:

$$\tilde{f}_s(t;\varepsilon) = \rho_s(t - \hat{y}_s;\varepsilon)$$

where

$$\rho_s(u;\varepsilon) := \begin{cases} p_s \cdot u & \text{if} \quad u \geq \varepsilon \\ -\frac{q_s-p_s}{4\varepsilon}u^2 + \frac{p_s+q_s}{2}u - \frac{(q_s-p_s)\varepsilon}{4} & \text{if} \quad -\varepsilon < u < \varepsilon \\ q_s \cdot u & \text{if} \quad u \leq -\varepsilon \ . \end{cases} \tag{B.1}$$

Indeed, note that for any positive ε, $\tilde{f}_s(t;\varepsilon)$ is an underestimation of $f_s(t)$. In fact, since $\tilde{f}_s(t;\varepsilon) = f_s(t)$ whenever $|t - \hat{y}_s| \geq \varepsilon$, it remains to consider the case when $|t| < \varepsilon$. When $u := t - \hat{y}_s \in [0, \varepsilon)$,

$$\begin{aligned} f_s(t) - \tilde{f}_s(t;\varepsilon) &= p_s u - \left(-\frac{q_s - p_s}{4\varepsilon}u^2 + \frac{p_s + q_s}{2}u - \frac{(q_s - p_s)\varepsilon}{4} \right) \\ &= \frac{q_s - p_s}{4\varepsilon}(u - \varepsilon)^2 \quad . \end{aligned}$$

Similarly, $f_s(t) - \tilde{f}_s(t;\varepsilon) = \dfrac{q_s - p_s}{4\varepsilon}(u + \varepsilon)^2$ when $u = t - \hat{y}_s \in (-\varepsilon, 0]$. This gives the quadratic spline in Section 4.1. Also note that $f_s(t) - \tilde{f}_s(t;\varepsilon) \in [0, (q_s - p_s)\varepsilon/4]$.

It follows that if $\bar{x}$ is optimal for the original piecewise linear problem (4.3) and $\tilde{x}$ is optimal for the approximate problem (4.4), then

$$\sum_{s=1}^{S} \pi_s f_s(r_s^T \bar{x}) - \sum_{s=1}^{S} \pi_s \tilde{f}_s(r_s^T \tilde{x}; \varepsilon) \leq \sum_{s=1}^{S} \pi_s f_s(r_s^T \bar{x}) - \sum_{s=1}^{S} \pi_s \bar{f}_s(r_s^T \bar{x}; \varepsilon) \leq \frac{(q_s - p_s)\varepsilon}{4} \quad .$$

Generally, given a convex-concave function $f(t) = f^+([t]^+) - f^-([t]^-)$ as in (2.3), where f^+ and f^- are assumed to be differentiable on $\mathbb{R}_{++}$, it is possible to make full use of the derivative information and use a cubic spline for approximation instead. Specifically, with resolution parameter $\varepsilon > 0$, f can be approximated by the

$$\tilde{f}(t;\varepsilon) := \begin{cases} f(t) & \text{if } |t| > \varepsilon \\ at^3 + bt^2 + ct + d & \text{if } |t| \leq \varepsilon \ , \end{cases}$$

where

$$\begin{aligned} a &= \frac{1}{2\varepsilon^2}\left(\frac{f'(\varepsilon) + f'(-\varepsilon)}{2} - \frac{f(\varepsilon) - f(-\varepsilon)}{2\varepsilon}\right) \\ b &= \frac{f'(\varepsilon) - f'(-\varepsilon)}{4\varepsilon} \\ c &= \frac{3\big(f(\varepsilon) - f(-\varepsilon)\big)}{4\varepsilon} - \frac{f'(\varepsilon) + f'(-\varepsilon)}{4} \\ d &= \frac{f(\varepsilon) + f(-\varepsilon)}{2} - \frac{\varepsilon\big(f'(\varepsilon) - f'(-\varepsilon)\big)}{4} \ . \end{aligned}$$