Robust Portfolio Selection Based on a Multi-stage Scenario Tree

Ruijun Shen * Shuzhong Zhang †

February 2006; revised November 2006

Abstract

The aim of this paper is to apply the concept of robust optimization introduced by Bel-Tal and Nemirovski to the portfolio selection problems based on multi-stage scenario trees. The objective of our portfolio selection is to maximize an expected utility function value (or equivalently, to minimize an expected disutility function value) as in a classical stochastic programming problem, except that we allow for ambiguities to exist in the probability distributions along the scenario tree. We show that such a problem can be formulated as a finite convex program in the conic form, on which general convex optimization techniques can be applied. In particular, if there is no short-selling, and the disutility function takes the form of semi-variance downside risk, and all the parameter ambiguity sets are ellipsoidal, then the problem becomes a second order cone program, thus tractable. We use SeDuMi to solve the resulting robust portfolio selection problem, and the simulation results show that the robust consideration helps to reduce the variability of the optimal values caused by the parameter ambiguity.

Keywords: portfolio selection, scenario tree, robust optimization, conic optimization.

MSC subject classification: 90C20, 90C99.

*Dillon Read Capital Management, Shanghai, China (rogerrjshen@yahoo.com.cn).

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong (zhang@se.cuhk.edu.hk). Research supported by Hong Kong RGC Earmarked Grants CUHK4174/03E and CUHK4242/04E.

1 Introduction

Although the mean-variance model has proved instructive, convenient, and useful for the theory and practice of portfolio selection, it is believed to be overly sensitive to parameter uncertainties and estimation errors. That is to say, a small perturbation in mean or variance can completely change the optimal portfolio. In the literature, a number of techniques have been developed to deal with this practical issue, among which we mention robust optimization, whose framework was established by Bel-Tal and Nemirovski [1]. Later, Goldfarb and Iyengar [7] applied the idea of robust optimization to mean-variance models, in which the parameters in the return vector and in the covariance matrix are assumed to be ambiguous. In this context, we make a distinction between *uncertainty* and *ambiguity* in an optimization model. By *uncertainty* we refer to the intrinsic stochastic nature of the model such as the return of a stock, while by *ambiguity* we refer to the limitation of our ability to localize the parameters of the model. Goldfarb and Iyengar [7] demonstrated that, by using a robust optimization approach to deal with the parameter ambiguity, the mean-variance portfolio selection problem, the maximum Sharpe ratio portfolio selection problem and the value-at-risk (VaR) portfolio selection problem can be reformulated as Second Order Cone Programs (SOCP), provided the parameter ambiguity sets are ellipsoidal. The use of minimax criteria within the framework of stochastic programming and risk management have been proposed previously by Breton and El Hachem [4] and Rustem and Howe [11]. Our emphasis, however, is the quest of how to immunize the scenario-tree based stochastic programming model from being overly sensitive to the the system parameters. Motivating examples on the issue of parameter ambiguity can be found in Ben-Tal and Nemirovski [2], albeit the model in question in [2] was simply linear programming. Preceding investigations along this direction include Dupačová [6], where however, the resulting optimization models were still hard to solve. The main contribution of the current paper is to propose a robust optimization resolution to stochastic programming based on a scenario tree, with the system parameters are modelled to reside in ellipsoidal ambiguity sets. The resulting eventual optimization models are shown to be in the realm of *Second Order Cone Programming*, which can be efficiently solved by, e.g., SeDeMi of Jos Sturm [13].

The paper is organized as follows. In Section 2 we introduce the investment problems and their corresponding mathematical programming models that are of interest to us. In particular, we introduce our single-stage and two-stage investment models and their robust optimization counter-parts. Then, in Section 3 we study the single-stage models in detail. We formulate a specific class of problems of (P_1) with no-short selling, ellipsoidal ambiguity sets and semi-variance disutility functions. (Note the negation of a disutility function is a utility function). This specific model will be written finally as a SOCP, which can be solved efficiently by existing optimization solvers such as SeDuMi [13]. Then we extend the model to a more general one. It can be shown that such a model, containing initially infinite constraints, can be transformed into a finite convex optimization problem in explicit form. Two-stage models are discussed in Section 4. A specific class of problems and a general class of problems are formulated respectively and the results are similar to those in the single-stage cases. Numerical results are presented in Section 5, followed by some concluding remarks.

2 The investment models

2.1 A single-stage utility maximizing investment model

Suppose we want to select a portfolio from n assets in the market and hold it for a given period. Without loss of generality, the initial wealth level is rescaled to 1. Suppose there are only m possible outcomes (or scenarios) of the market at the end of the holding period; that is, all possible scenarios can be described by a single-stage tree with m leaves. In our investment problem, different scenarios are characterized by the different returns of assets. Since there are totally n assets, an $n \times 1$ vector is sufficient to contain all information. Here we assume the i^{th} element of the return vector represents the return of i^{th} asset. Note a "return" mentioned above includes both gain/loss and principal. For example, an increase of 10% in a single stock will be recorded as a return of 110%. If, by statistical analysis or past experiences, such return vectors and the probabilities of the occurrences of the scenarios are already known, then the single-stage portfolio selection model based on the scenario tree can be described as

$$\begin{array}{lll} (P_1) & \max & \sum_{i=1}^{m} \pi_i u(\phi^T r^i) \\ & \text{s.t.} & \phi^T e = 1 \\ & & \phi \in \Delta, \end{array}$$

where

n:	the number of stocks;
m:	the number of subsequent scenarios at each node;
$\phi \in \Re^n$:	the holding of stocks, which is the decision vector in the model;
$r^i \in \Re^n$:	the return of n stocks if scenario i occurs;
π_i:	the probability for scenario i to occur;
$e \in \Re^n$:	the vector of all 1's;
Δ:	the set of admissible portfolios, which is assumed to be convex.

2.2 A two-stage utility maximizing investment model

The single-stage model can be extended to the multi-stage ones. For simplicity, we consider a perfect two-stage scenario tree, in the sense that all leaf nodes are of the same depth and all the internal nodes have degree m. Thus, the two-stage portfolio selection model based on this scenario tree can be written as

$$\begin{array}{ll} \max & \sum_{i=1}^{m} \pi_i \left[\sum_{j=1}^{m} \pi_j^i u(\phi^{i*^T} r^{ij}) \right] \\ \text{s.t.} & \phi^T e = 1 \\ & \phi \in \Delta, \end{array}$$

where

n:	the number of stocks;
m:	the number of subsequent scenarios at each node;
$\phi \in \Re^n$:	the holding of stocks at first stage;
$r^i \in \Re^n$:	the return of n stocks if scenario i occurs;
$r^{ij} \in \Re^n$:	the return of n stocks if scenario i occurs at the first stage and scenario j occurs at the second stage;
π_i:	the probability that scenario i occurs at the first stage;
π_j^i:	the conditional probability given that scenario j occurs at the second stage given that scenario i occurs at the first stage;
$e \in \Re^n$:	the vector of all 1's;
Δ:	the set of admissible portfolios at first stage;
Δ^i:	the set of admissible portfolios at second stage.

Finally, $\phi^{i*} \in \Re^n$ is the optimal solution to the second-stage recourse problem:

$$\begin{array}{ll} \max & \sum_{j=1}^{m} \pi_j^i u(\phi^{i^T} r^{ij}) \\ \text{s.t.} & \phi^{i^T} e = \phi^T r^i, \forall i = 1, 2, ..., m \\ & \phi^i \in \Delta^i, \end{array}$$

where $\phi^i \in \Re^n$ is the holding of stocks at the second stage.

To make our presentation concise, a multi-stage optimization problem in the above form will be written in the following compact format, provided that there is no confusion:

$$\begin{array}{llll} (P_2) & \max\limits_{\phi} & \sum_{i=1}^{m} \pi_i & \begin{array}{ll} \max\limits_{\phi^i} & \sum_{j=1}^{m} \pi_j^i u(\phi^{i^T} r^{ij}) \\ \text{s.t.} & \phi^{i^T} e = \phi^T r^i, \forall i = 1, 2, ..., m \\ & \phi^i \in \Delta^i \end{array} \\ & \text{s.t.} & \phi^T e = 1 & \\ & & \phi \in \Delta. & \end{array}$$

Some explanations are in order here.

1. ϕ, ϕ^i are vectors, the k^{th} element of which represents the value invested in k^{th} asset.

2. the sets of admissible portfolios represent all conditions other than budget constraints. A typical example is no short-selling, which corresponds to $\Delta = \Delta^i = \Re^n_+$. If there are no other constraints, the sets are simply $\Re^n$. To make (P_2) a convex problem, we assume the admissible sets are convex.

We set $\pi = (\pi_1, \cdots, \pi_m)^T$ and $\pi^i = \left(\pi^i_1, \cdots, \pi^i_m\right)^T$. By definition of probabilities, π and π^i's are nonnegative vectors with $\pi^T e = 1$ and ${\pi^i}^T e = 1$.

Clearly, (P_2) is separable and hence has the same solution as

$$\begin{array}{ll} \max\limits_{\phi,\phi^i} & \sum\limits_{i=1}^{m} \pi_i \left[\sum\limits_{j=1}^{m} \pi^i_j u({\phi^i}^T r^{ij}) \right] \\ \text{s.t.} & {\phi^i}^T e = \phi^T r^i, \forall i = 1, 2, ..., m \\ & \phi^i \in \Delta^i, \phi^T e = 1, \phi \in \Delta. \end{array}$$

It is easy to see that the above problem is a convex program, since $u(\cdot)$ is concave. Such problems are classical and have been well addressed in the literature.

2.3 Robust counterpart of the single-stage model

In (P_1) a deterministic scenario tree is assumed. One drawback of this approach is that the scenario tree generated may not be accurate. Basically, there are two kinds of ambiguities that may occur in a scenario tree: the ambiguity in the returns in each scenario and the ambiguity in conditional probabilities that each scenario will happen. In fact, what we use in (P_1) are the estimated vectors $\tilde{\pi}, \tilde{r}^1, ..., \tilde{r}^m$ instead of their actual realizations $\pi, r^1, ..., r^m$. Note the estimated probabilities should also satisfy $\tilde{\pi}^T e = 1$ by definition.

Since scenario trees are usually generated by estimations, we are not sure what will be the real probabilities and returns. However, we do have some confidence that the actual values are not far away from the estimated ones, and by statistical analysis, we can possibly obtain some confidence intervals. Mathematically, we assume that the return vectors and probability vectors lie in some ambiguity sets containing the corresponding estimated values:

- $r^i \in V^i$ (ambiguity in returns),
- $\pi \in \Pi$ (ambiguity in probability distribution).

We assume that all the above sets are convex and compact. Moreover, these sets are non-empty.

We can simplify the probability ambiguity set as follows. Let $y = \pi - \tilde{\pi}, U = \Pi - \tilde{\pi}$. Then

$$\pi \in \Pi \Longleftrightarrow y \in U.$$

Clearly, the so-constructed set U remains compact and convex. It is easy to see, according to Bel-Tal and Nemirovski's framework, the robust counterpart of the single-stage model is

$$
\begin{array}{lll}
(RP_1) & \max\limits_{\phi} & \min\limits_{r^i \in V^i, y \in U} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) u(\phi^T r^i) \\
 & \text{s.t.} & \phi^T e = 1 \\
 & & \phi \in \Delta.
\end{array}
$$

2.4 Robust counterpart of the two-stage model

Similarly, we can describe the uncertainties in a two-stage tree. Suppose that what we used in (P_2) are the estimated vectors $\tilde{\pi}, \tilde{r}^i, \tilde{\pi}^i, \tilde{r}^{ij}$, while their actual realizations are π, r^i, π^i, r^{ij} respectively. Note that the estimated probabilities should satisfy $\tilde{\pi}^T e = 1$ and $\tilde{\pi}^{i^T} e = 1$ also.

We assume that the actual return vectors and probability vectors lie in the following ambiguity sets containing the corresponding estimated values:

- (ambiguity in the return vectors) $r^i \in V^i$ and $r^{ij} \in V^{ij}$;
- (ambiguity in probability distribution) $\pi \in \Pi$ and $\pi^i \in \Pi^i$.

We assume that all the above sets are convex, compact and non-empty. Moreover, we assume that $\tilde{\pi}, \tilde{r}^i, \tilde{\pi}^i, \tilde{r}^{ij}$ are independent from each other.

We can simplify the probability ambiguity sets as follows. Let $y = \pi - \tilde{\pi}, U = \Pi - \tilde{\pi}$. Then $\pi \in \Pi \iff y \in U$. Similarly, let $y^i = \pi^i - \tilde{\pi}^i, U^i = \Pi^i - \tilde{\pi}^i$. Then $\pi \in \Pi \iff y^i \in U^i$. Clearly, U, U^i so constructed are still compact convex sets.

Bel-Tal and Nemirovski's framework of robust optimization is for single-period problems. Following the same general principle, the model can certainly be extended to the multiple stage case. That is to say, when we choose an initial portfolio, the worst possible π and r^is are assumed to have unfolded at the end of the first period, and for given ϕ, r^i in each recourse problem, if we switch our portfolio to ϕ^i at the beginning of the second period, then again the worst scenario π^i and r^{ij} will unfold. Therefore we define the robust counterpart of (P_2) as:

$$
\begin{array}{llllll}
(RP_2) & \max\limits_{\phi} & \min\limits_{r^i \in V^i, y \in U} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) & \max\limits_{\phi^i} & \min\limits_{r^{ij} \in V^{ij}, y^i \in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) u(\phi^{i^T} r^{ij}) \\
 & & & \text{s.t.} & \phi^{i^T} e = \phi^T r^i, \forall i = 1, 2, ..., m \\
 & & & & \phi^i \in \Delta^i \\
 & \text{s.t.} & \phi^T e = 1 & & \\
 & & \phi \in \Delta. & &
\end{array}
$$

3 Finite representation of the single-stage robust model

3.1 A downside minimization model

To start with, we consider a specific portfolio selection problem in the class of (P_1) satisfying the following assumptions.

3.1.1 Assumptions

- There is no short selling: $\Delta = \Re^n_+$.
- We face a semi-variance disutility function $d(w) = (R-w)^2_+$. Such a disutility function represents the downside risks with respect to a given benchmark return R. This means that the corresponding utility function is $u(w) = -(R-w)^2_+$.
- All assets we consider are financial securities, and as a result an investor can not lose more than his initial wealth, namely $V^i \subseteq \Re^n_+$.
- The ambiguity sets are ellipsoids in the following forms:

$$\begin{array}{l} \Pi = \{\pi \in \Re^m \mid \pi^T e = 1, \|\pi - \tilde{\pi}\| \leq \theta\}, \\ V^i = \{r^i \in \Re^n \mid (r^i - \tilde{r}^i)^T Q^i (r^i - \tilde{r}^i) \leq \rho_i^2\}, i = 1, 2, ..., m. \end{array}$$

  Using simplifications in the previous section,

$$U = \{y \in \Re^m \mid y^T e = 0, \|y\| \leq \theta\}.$$

For simplicity we assume all Q^i's are identity matrices. As a result, our ambiguity sets become

$$\begin{array}{l} U = \{y \in \Re^m \mid y^T e = 0, \|y\| \leq \theta\}, \\ V^i = \{r^i \in \Re^n \mid \|r^i - \tilde{r}^i\| \leq \rho_i\}. \end{array}$$

3.1.2 Formulation of the model

With the above assumptions, the original problem can now be formulated as

$$\begin{array}{lll} (SP_1) & \max\limits_{\phi} & \sum\limits_{i=1}^{m} \tilde{\pi}_i \left[-(R - \phi^T \tilde{r}^i)^2_+\right] \\ & \text{s.t.} & \phi^T e = 1 \\ & & \phi \geq 0. \end{array}$$

Obviously, the robust counterpart of (SP_1) is

$$\begin{array}{lll} (RSP_1) & \max\limits_{\phi} & \min\limits_{r^i \in V^i, y \in U} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) \left[-(R - \phi^T r^i)_+^2 \right] \\ & \text{s.t.} & \phi^T e = 1 \\ & & \phi \geq 0. \end{array}$$

3.1.3 Solution for the model

(RSP_1) can be equivalently put as

$$\begin{array}{ll} \min\limits_{\phi, t_i, t_0} & t_0 \\ \text{s.t.} & t_0 \geq \max\limits_{y \in U} (\tilde{\pi} + y)^T t \\ & t_i \geq \max\limits_{r^i \in V^i} (R - \phi^T r^i)_+^2 \\ & \phi^T e = 1, \, \phi \geq 0. \end{array}$$

The key is to formulate the model in a tractable form. To this end, we introduce the notion of *Second Order Cone Programming* (SOCP). A standard second order cone in $\Re^{k+1}$ is defined as

$$\text{SOC } (k+1) = \left\{ \left(\begin{array}{c} t_0 \\ t \end{array} \right) \in \Re^{k+1} \,\middle|\, t_0 \in \Re, \, t \in \Re^k, \, t_0 \geq \|t\| \right\}.$$

Clearly, SOC $(k+1)$ is a closed convex cone. A general second order cone is a Cartesian product of a finite number of standard second order cones, and a second order cone program is to optimize a linear objective function over the intersection of an affine subspace and a general second order cone. Such problems can be solved efficiently by Interior Point Methods. For more information on SOCP and other conic optimization models, one is referred to an excellent text by Ben-Tal and Nemirovski [3] or the Ph.D. thesis of Jos Sturm [14], which contains a through treatment on the duality theory of SOCP and SDP. The SOCP models can be efficiently solved by SeDuMi of Jos Sturm [13] or MOSEK[1]. It is well known that, although SOCP is much more general than LP (Linear Programming), its computational tractability is comparable to that of LP, in the face of Primal-Dual Interior Point Methods. Our goal now is to formulate (RSP_1) in the form of SOCP. First, observe the following simple fact, which can be verified using the KKT optimality condition.

Lemma 3.1 *The optimal solution to* $\min\limits_{y \in U} -a^T y$ *is*

$$y^* = \frac{\theta \left(a - \frac{e^T a}{m} e \right)}{\sqrt{a^T a - \frac{(e^T a)^2}{m}}} \textit{ with } a^T y^* = \theta \sqrt{a^T a - \frac{(e^T a)^2}{m}} = \theta \|a - e \frac{a^T e}{m}\|.$$

[1]http://www.mosek.com/

It follows from the above lemma that:

Corollary 3.2 $t_0 \geq \sum_{i=1}^{m} (\tilde{\pi}_i + y_i) a_i, \forall y \in U$ *is equivalent to*

$$\begin{pmatrix} t_0 - \tilde{\pi}^T a \\ \theta(a - e\frac{a^T e}{m}) \end{pmatrix} \in \text{SOC } (m+1).$$

We now add some more variables and reformulate (RSP_1) as

$$\begin{array}{ll} \min\limits_{\phi, t_i, t_0} & t_0 \\ \text{s.t.} & t_0 \geq \max\limits_{y \in U} (\tilde{\pi} + y)^T t \\ & t_i \geq \tau_i^2 \\ & \tau_i \geq \max\limits_{r^i \in V^i} (R - \phi^T \tilde{r}^i) \\ & \tau_i \geq 0 \\ & \phi^T e = 1 \\ & \phi \geq 0. \end{array}$$

By Corollary 3.2,

$$t_0 \geq \sum_{i=1}^{m} (\tilde{\pi}_i + y_i) t_i, \forall y \in U \Longleftrightarrow \begin{pmatrix} t_0 - \tilde{\pi}^T t \\ \theta[e \cdot \frac{t^T e}{m} - t] \end{pmatrix} \in \text{SOC } (m+1),$$

For a given ϕ,

$$\max_{r^i} (R - \phi^T r^i) = R - \phi^T \tilde{r}^i + \rho_i \|\phi\|.$$

Also

$$t_i \geq \tau_i^2 \Longleftrightarrow \begin{pmatrix} t_i + 1 \\ t_i - 1 \\ 2\tau_i \end{pmatrix} \in \text{SOC } (3).$$

Hence (RSP_1) can be reformulated into the following Second Order Cone Program (SOCP):

$$\begin{array}{ll} \min\limits_{\phi, \tau_i, t_i, t_0} & t_0 \\ \text{s.t.} & \begin{pmatrix} t_0 - \tilde{\pi}^T t \\ \theta(e \cdot \frac{t^T e}{m} - t) \end{pmatrix} \in \text{SOC } (m+1), \begin{pmatrix} t_i + 1 \\ t_i - 1 \\ 2\tau_i \end{pmatrix} \in \text{SOC } (3) \\ & \begin{pmatrix} \tau_i - R + \phi^T \tilde{r}^i \\ \rho_i \phi \end{pmatrix} \in \text{SOC } (n+1) \\ & \tau_i \geq 0, \phi \geq 0, \phi^T e = 1. \end{array} \tag{1}$$

Although the uncertain sets we are considering are ellipsoidal, it is easy to arrive at a similar SOCP when the uncertain sets are the intersections of finite number of ellipsoids. Besides, the specification of Q^is also increases the flexibility of model (RSP_1). For example, if asset 1 is risk-free, then we can set

$$Q^i = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}, \forall i$$

to remove the uncertainties on its returns in all the scenarios.

3.2 The general model

We proceed to solve problem (RP_1) in the general form:

$$\begin{array}{lll} (RP_1) & \max\limits_{\phi} & \min\limits_{r^i \in V^i, y \in U} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) u(\phi^T r^i) \\ & \text{s.t.} & \phi^T e = 1 \\ & & \phi \in \Delta. \end{array}$$

By adding new variables, (RP_1) becomes

$$\begin{array}{ll} \max\limits_{\phi} & t_0 \\ \text{s.t.} & t_0 \leq \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) u_i, \forall y \in U \\ & u_i \leq u(w_i) \\ & w_i \leq \phi^T r^i, \forall r^i \in V^i \\ & \phi^T e = 1 \\ & \phi \in \Delta, \end{array}$$

since u is increasing.

There are still infinite amount of constraints in the above representation. Fortunately, there is an elegant dual structure to help us. Before we proceed, it is necessary to introduce some fundamental concepts in duality theory (see [15] and [10] for reference).

Definition 3.3 *If D is a convex set, then its homogenized cone is*

$$\text{H}(D) = \text{cl} \left\{ \begin{pmatrix} t \\ x \end{pmatrix} \middle| t > 0, \frac{x}{t} \in D \right\}.$$

Definition 3.4 *Let K be a cone. The dual cone of K is*

$$K^* = \{x \mid \langle x, y \rangle \geq 0, \forall y \in K\}.$$

For a given $k \times n$ matrix B and a cone K:

$$Bx \in K^* \Longleftrightarrow y^T Bx \geq 0,\ \forall y \in K \Longleftrightarrow x \in \{B^T y \mid y \in K\}^*.$$

Now we can reformulate infinite constraints by finite representations in the form of dual cones. Setting $\mathbf{t} = (t_1, \cdots, t_m)^T$,

$$t_0 \leq \sum_{i=1}^{m} (\tilde{\pi}_i + y_i) t_i, \forall y \in U$$

is equivalent to

$$\left\langle \begin{pmatrix} \tilde{\pi}^T \mathbf{t} - t_0 \\ \mathbf{t} \end{pmatrix}, \begin{pmatrix} y_0 \\ y \end{pmatrix} \right\rangle \in \Re_+, \forall \begin{pmatrix} y_0 \\ y \end{pmatrix} \in \mathsf{H}(U).$$

This, in the dual form, is simply

$$\begin{pmatrix} \tilde{\pi}^T \mathbf{t} - t_0 \\ \mathbf{t} \end{pmatrix} \in \mathsf{H}(U)^*.$$

We can apply similar transformations on $w_i \geq \phi^T r^i$ and obtain the following finite convex representation of (RP_1):

$$\begin{array}{ll} \min\limits_{\phi} & -t_0 \\ \text{s.t.} & \begin{pmatrix} \tilde{\pi}^T \mathbf{t} - t_0 \\ \mathbf{t} \end{pmatrix} \in \mathsf{H}(U)^*,\ u_i \leq u(w_i),\ \begin{pmatrix} -w_i \\ \phi \end{pmatrix} \in \mathsf{H}(V^i)^* \\ & \phi^T e = 1, \phi \in \Delta. \end{array} \tag{2}$$

4 Finite representation of the two-stage robust model

4.1 The downside minimization model revisited

4.1.1 Assumptions

It is natural to extend the single-stage models to the multi-stage cases. Like in the single-stage case, we start with a specific two-stage portfolio selection problem based on the following assumptions:

- There is no short selling: $\Delta = \Delta^i = \Re^n_+$.
- We use a semi-variance disutility function

$$d(w) = (R - w)^2_+.$$

- All assets that we consider are stocks and as a result, an investor can not have negative wealth:

$$V^i, V^{ij} \subseteq \Re^n_+.$$

- The ambiguity sets are ellipsoids in the following forms:

$$\begin{array}{l}\Pi = \{\pi \in \Re^m \mid \pi^T e = 1, \|\pi - \tilde{\pi}\| \leq \theta\} \\ V^i = \{r^i \in \Re^n \mid (r^i - \tilde{r}^i)^T Q^i (r^i - \tilde{r}^i) \leq \rho_i^2\} \\ \Pi^i = \{\pi \in \Re^m \mid \pi^{i^T} e = 1, \|\pi^i - \tilde{\pi}^i\| \leq \theta_i\}, i = 1, 2, ..., m \\ V^{ij} = \{r^{ij} \in \Re^n \mid (r^{ij} - \tilde{r}^{ij})^T Q^{ij} (r^{ij} - \tilde{r}^{ij}) \leq \rho_{ij}^2\}, i = 1, 2, ..., m, j = 1, 2, ..., m.\end{array}$$

  Using simplifications as in the previous section,

$$\begin{array}{l}U = \{y \in \Re^m \mid y^T e = 0, \|y\| \leq \theta\} \\ U^i = \{y^i \mid y^{i^T} e = 0, \|y^i\| \leq \theta^i\}, i = 1, 2, ..., m.\end{array}$$

  We assume w.l.o.g that all Q^i, Q^{ij}'s are identity matrices. Consequently, our ambiguity sets become

$$\begin{array}{l}U = \{y \in \Re^m \mid y^T e = 0, \|y\| \leq \theta\}, \\ V^i = \{r^i \in \Re^n \mid \|(r^i - \tilde{r}^i)\| \leq \rho_i\}, \\ U^i = \{y^i \in \Re^m \mid y^{i^T} e = 0, \|y^i\| \leq \theta_i\}, \\ V^{ij} = \{r^{ij} \in \Re^n \mid \|(r^{ij} - \tilde{r}^{ij})\| \leq \rho_{ij}\}.\end{array}$$

4.1.2 Formulation of the model

Under assumptions we have made in the last section, the specific robust optimization problem that is of interest to us is

$$\begin{array}{llllll}(RSP_2) & \max\limits_{\phi} & \min\limits_{r^i, y} \sum\limits_{i=1}^m (\tilde{\pi}_i + y_i) & \max\limits_{\phi^i} & \min\limits_{r^{ij}, y^i} \sum\limits_{j=1}^m (\tilde{\pi}^i_j + y^i_j)[-(R - \phi^{i^T} r^{ij})^2_+] \\ & & & \text{s.t.} & \phi^{i^T} e = \phi^T r^i, \forall i = 1, 2, ..., m \\ & & & & \phi^i \geq 0 \\ & \text{s.t.} & \phi^T e = 1 \\ & & \phi \geq 0. \end{array}$$

Obviously, this is an extension of (RSP_1). Again, the negative sign of the utility function can be removed by swapping the 'min' and 'max' operations throughout, leading to an equivalent problem with the same optimal solutions:

$$\begin{array}{lllll}\min\limits_{\phi} & \max\limits_{r^i, y} \sum\limits_{i=1}^m (\tilde{\pi}_i + y_i) & \min\limits_{\phi^i} & \max\limits_{r^{ij}, y^i} \sum\limits_{j=1}^m (\tilde{\pi}^i_j + y^i_j)(R - \phi^{i^T} r^{ij})^2_+ \\ & & \text{s.t.} & \phi^{i^T} e = \phi^T r^i, \forall i = 1, 2, ..., m \\ & & & \phi^i \geq 0 \\ \text{s.t.} & \phi^T e = 1 \\ & \phi \geq 0. \end{array} \tag{3}$$

Note that (3) is a special case of

$$
\begin{array}{llll}
\min\limits_{\phi} & \max\limits_{r^i,y} \sum\limits_{i=1}^{m}(\tilde{\pi}_i + y_i) & \min\limits_{\phi^i} & \max\limits_{r^{ij},y^i} \sum\limits_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T r^{ij})^2_+ \\
 & & \text{s.t.} & {\phi^i}^T e = \phi^T r^i, \forall i = 1,2,...,m \\
 & & & \phi^i \geq 0 \\
\text{s.t.} & \phi^T e = 1 & & \\
 & \phi \geq 0. & &
\end{array}
\tag{4}
$$

Actually, the above is the problem we are going to solve.

4.1.3 Solution for the model

For a given ϕ^i,

$$\max_{r^{ij}\in V^{ij}} (R^{ij} - {\phi^i}^T r^{ij}) = R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \rho_{ij}\|\phi^i\|.$$

The outcome of r^i in the first stage affects the second stage disutility only through $\phi^T r^i$, and it is easy to see that

$$\min_{r^i\in V^i} \phi^T r^i = \phi^T \tilde{r}^i - \rho_i\|\phi\|,$$

and

$$\max_{r^i\in V^i} \phi^T r^i = \phi^T \tilde{r}^i + \rho_i\|\phi\|.$$

So $w_i : r^i \to \phi^T r^i$, maps the ball V^i onto the interval $[\phi^T\tilde{r}^i - \rho_i\|\phi\|, \phi^T\tilde{r}^i + \rho_i\|\phi\|]$. In fact, this mapping is continuous and surjective. This means that (4) is equivalent to

$$
\begin{array}{lllll}
\min\limits_{\phi} & \max\limits_{y} \sum\limits_{i=1}^{m}(\tilde{\pi}_i + y_i) & \max\limits_{w_i} & \min\limits_{\phi^i} & \max\limits_{y^i} \sum\limits_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \|\phi^i\|)^2_+ \\
 & & & \text{s.t.} & {\phi^i}^T e = w_i, \forall i = 1,2,...,m \\
 & & & & \phi^i \geq 0 \\
 & & \text{s.t.} & & \phi^T\tilde{r}^i - \rho_i\|\phi\| \leq w_i \leq \phi^T\tilde{r}^i + \rho_i\|\phi\| \\
\text{s.t.} & \phi^T e = 1 & & & \\
 & \phi \geq 0. & & &
\end{array}
\tag{5}
$$

Note $w_i \geq 0$, due to the no short-selling assumption.

Lemma 4.1 *If $w_i \geq 0$, then*

$$
\begin{array}{ll}
\min\limits_{\phi^i} & \max\limits_{y^i} \sum\limits_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \|\phi^i\|)^2_+ \\
\text{s.t.} & {\phi^i}^T e = w_i \\
 & \phi^i \geq 0
\end{array}
\tag{6}
$$

and

$$\begin{array}{lll} \min\limits_{\phi^i} & \max\limits_{y^i} \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \|\phi^i\|)^2_+ & \\ \text{s.t.} & {\phi^i}^T e \leq w_i & \\ & \phi^i \geq 0 & \end{array} \tag{7}$$

are both solvable and have the same optimal value.

Proof: First, $\phi_i = (w_i, 0, \cdots, 0)^T$ is feasible to both problems. Since the semi-variance disutility is non-negative, both problems are bounded below by 0. Because the constraint sets are compact and the objective function

$$\max_{y^i} \sum_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \|\phi^i\|)^2_+$$

in both problems are continuous in ϕ^i, the minimum values can always be attained. If $\phi^{i^*}_a$ is optimal to (6), then it must be feasible to (7), which means the optimal value of (6) is larger than or equals to that of (7). Conversely, if $\phi^{i^*}_b$ is optimal to (7), then we can take $\phi^i_a = (w_i - {\phi^{i^*}_b}^T e)e_1 + \phi^{i^*}_b$, which is clearly feasible to (6). According to our assumption,

$$\begin{array}{ll} & {\phi^{i}_a}^T r^{ij} \geq {\phi^{i*}_b}^T r^{ij}, \forall r^{ij} \in V^{ij} \\ \Longrightarrow & \max\limits_{r^{ij} \in V^{ij}} {\phi^{i}_a}^T r^{ij} \geq \max\limits_{r^{ij} \in V^{ij}} {\phi^{i*}_b}^T r^{ij} \\ \Longrightarrow & R^{ij} - {\phi^{i}_a}^T \tilde{r}^{ij} + \|\phi^i_a\| \leq R^{ij} - {\phi^{i*}_b}^T \tilde{r}^{ij} + \|\phi^{i*}_b\| \\ \Longrightarrow & \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^{i}_a}^T \tilde{r}^{ij} + \|\phi^i_a\|)^2_+ \leq \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^{i*}_b}^T \tilde{r}^{ij} + \|\phi^{i*}_b\|)^2_+, \forall y^i \in U^i \\ \Longrightarrow & \max\limits_{y^i \in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^{i}_a}^T \tilde{r}^{ij} + \|\phi^i_a\|)^2_+ \leq \max\limits_{y^i \in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^{i*}_b}^T \tilde{r}^{ij} + \|\phi^{i*}_b\|)^2_+. \end{array}$$

Hence (6) has an optimal value less than or equal to that of (6). We conclude that (6) and (7) have the same optimal value. □

Therefore, (5) has the same solution as

$$\begin{array}{lllll} \min\limits_{\phi} & \max\limits_{y} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) & \max\limits_{w_i} & \min\limits_{\phi^i} & \max\limits_{y^i} \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \|\phi^i\|)^2_+ \\ & & & \text{s.t.} & {\phi^i}^T e \leq w_i, \forall i = 1, 2, ..., m \\ & & & & \phi^i \geq 0 \\ & & \text{s.t.} & & \phi^T \tilde{r}^i - \rho_i \|\phi\| \leq w_i \leq \phi^T \tilde{r}^i + \rho_i \|\phi\| \\ \text{s.t.} & \phi^T e = 1 & & & \\ & \phi \geq 0. & & & \end{array} \tag{8}$$

Clearly, the smaller the w_i, the smaller the constraint sets of (7). Therefore, (8) must have the same optimal value as

$$\begin{array}{llll} \min\limits_{\phi} & \max\limits_{y} \sum\limits_{i=1}^{m}(\tilde{\pi}_i + y_i) & \min\limits_{\phi^i} & \max\limits_{y^i} \sum\limits_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \|\phi^i\|)^2_+ \\ & & \text{s.t.} & {\phi^i}^T e \le \phi^T \tilde{r}^i - \rho_i \|\phi\|, \forall i = 1, 2, ..., m \\ & & & \phi^i \ge 0 \\ \text{s.t.} & \phi^T e = 1 & & \\ & \phi \ge 0. & & \end{array} \tag{9}$$

By the minimax theorem (cf. [12]), the first 'max' and the second 'min' in the objective function in (9) can be interchanged because the concerning function is convex in ϕ^i's and affine (therefore concave) in y.

This way, (9) can be transformed into

$$\begin{array}{ll} \min\limits_{\phi,\phi^i} & \max\limits_{y} \sum\limits_{i=1}^{m}(\tilde{\pi}_i + y_i) \max\limits_{y^i} \sum\limits_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)(R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \rho_{ij}\|\phi^i\|)^2_+ \\ \text{s.t.} & \rho_i \|\phi\| \le \phi^T \tilde{r}^i - {\phi^i}^T e, \forall i = 1, 2, ..., m \\ & \phi^i \ge 0, \phi^T e = 1, \phi \ge 0. \end{array} \tag{10}$$

By adding new variables, (10) can be further reformulated as

$$\begin{array}{ll} \min\limits_{\phi,\phi^i,u_{ij},t_i,t_0} & t_0 \\ \text{s.t.} & t_0 \ge \sum\limits_{i=1}^{m}(\tilde{\pi}_i + y_i)t_i, \forall y \in U \\ & t_i \ge \sum\limits_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)d_{ij}, \forall y^i \in U^i \\ & d_{ij} \ge (R^{ij} - {\phi^i}^T \tilde{r}^{ij} + \rho_{ij}\|\phi^i\|)^2_+ \\ & {\phi^i}^T e \le \phi^T \tilde{r}^i - \rho_i \|\phi\|, \forall i = 1, 2, ..., m \\ & \phi^i \ge 0, \phi^T e = 1, \phi \ge 0. \end{array} \tag{11}$$

We set $\mathbf{d_i} = \begin{pmatrix} d_{i1} \\ \vdots \\ d_{im} \end{pmatrix}$. By Corollary 3.2

$$t_0 \ge \sum_{i=1}^{m}(\tilde{\pi}_i + y_i)t_i, \forall y \in U \Leftrightarrow \begin{pmatrix} t_0 - \tilde{\pi}^T \mathbf{t} \\ \theta[e \cdot \frac{\mathbf{t}^T e}{m} - \mathbf{t}] \end{pmatrix} \in \text{SOC } (m+1),$$

and

$$t_i \ge \sum_{j=1}^{m}(\tilde{\pi}^i_j + y^i_j)d_{ij}, \forall y^i \in U^i \Leftrightarrow \begin{pmatrix} t_i - {\tilde{\pi}^i}{}^T \mathbf{d_i} \\ \theta_i[e \cdot \frac{\mathbf{d_i}^T e}{m} - \mathbf{d_i}] \end{pmatrix} \in \text{SOC } (m+1).$$

Also, the condition

$$d_{ij} \geq (R^{ij} - \phi^{i^T} \tilde{r}^{ij} + \rho_{ij} \|\phi^i\|)_+^2$$

is equivalent to

$$\begin{array}{l} d_{ij} \geq \tau_{ij}^2 \\ \tau_{ij} \geq R^{ij} - \phi^{i^T} \tilde{r}^{ij} + \rho_{ij} \|\phi^i\| \ , \\ \tau_{ij} \geq 0 \end{array}$$

i.e.

$$\begin{pmatrix} d_{ij} + 1 \\ d_{ij} - 1 \\ 2\tau_{ij} \end{pmatrix} \in \text{SOC } (m+1)$$

$$\begin{array}{l} \begin{pmatrix} \tau_{ij} - R^{ij} + \phi^{i^T} \tilde{r}^{ij} \\ \rho_{ij} \phi^i \end{pmatrix} \in \text{SOC } (m+1) \\ \tau_{ij} \geq 0. \end{array}$$

Consequently, (11) can be transformed into the following SOCP:

$$\begin{array}{cl} \min\limits_{\phi, \phi^i, d_{ij}, t_i, t_0} & t_0 \\ \text{s.t.} & \begin{pmatrix} t_0 - \tilde{\pi}^T \mathbf{t} \\ \theta[e \cdot \frac{\mathbf{t}^T e}{m} - \mathbf{t}] \end{pmatrix} \in \text{SOC } (m+1) \\ & \begin{pmatrix} t_i - \tilde{\pi}^{i^T} \mathbf{d_i} \\ \theta_i[e \cdot \frac{\mathbf{d_i}^T e}{m} - \mathbf{d_i}] \end{pmatrix} \in \text{SOC } (m+1), \forall i = 1, 2, ..., m \\ & \begin{pmatrix} \phi^T r^i - \phi^{i^T} e \\ \rho_i \phi \end{pmatrix} \in \text{SOC } (n+1), \forall i = 1, 2, ..., m \\ & \begin{pmatrix} d_{ij} + 1 \\ d_{ij} - 1 \\ 2\tau_{ij} \end{pmatrix} \in \text{SOC } (3), \begin{pmatrix} \tau_{ij} - R^{ij} + \phi^{i^T} \tilde{r}^{ij} \\ \rho_{ij} \phi^i \end{pmatrix} \in \text{SOC } (n+1) \\ & \tau_{ij} \geq 0, \phi^i \geq 0, \phi^T e = 1, \phi \geq 0. \end{array} \tag{12}$$

Obviously, the feasible region will be enlarged with the decrease of parameters $\theta, \theta_i, \rho_i, \rho_{ij}$. If $\theta, \theta_i, \rho_i, \rho_{ij}$ are all set to be zeros, then (12) reduces to

$$\begin{array}{ll}
\min\limits_{\phi,\phi^i,d_{ij},t_i,t_0} & t_0 \\
\text{s.t.} & t_0 - \tilde{\pi}^T \mathbf{t} \geq 0 \\
& t_i - \tilde{\pi}^{i^T} \mathbf{d_i} \geq 0, \forall i = 1, 2, ..., m \\
& \phi^T r^i - \phi^{i^T} e \geq 0, \forall i = 1, 2, ..., m \\
& \begin{pmatrix} d_{ij} + 1 \\ d_{ij} - 1 \\ 2\tau_{ij} \end{pmatrix} \in \text{SOC } (3) \\
& \tau_{ij} - R^{ij} + \phi^{i^T} \tilde{r}^{ij} \geq 0 \\
& \tau_{ij} \geq 0, \phi^i \geq 0, \phi^T e = 1, \phi \geq 0,
\end{array}$$

which is consistent with the formulation of the problem without uncertainties.

4.2 The general two-stage robust model

In this section we are going to solve the general problem (RP_2) proposed at the end of the introduction part. Before moving on, we make the following mild assumptions.

4.2.1 Assumptions

1. $u(\cdot)$ is increasing, continuous and concave;
2. U, U^i, V^i, V^{ij} are convex sets;
3. The first asset is a risk-free asset with positive return;
4. Δ is a convex set such that
   (a) $\phi \in \Delta \Longrightarrow \phi + te_1 \in \Delta, \forall t \geq 0$;
   (b) The set $\{\phi \mid \phi^T e = 1, \phi \in \Delta\}$ is compact;
5. For each i, Δ^i is a convex set such that
   (a) $\phi^i \in \Delta^i \Longrightarrow \phi^i + te_1 \in \Delta^i, \forall t \geq 0$;
   (b) The set $\{\phi^i | \phi^{i^T} e \leq w, \phi^i \in \Delta^i\}$ is compact for any $w \in \Re$.

Note

$$\{\phi^i \mid \phi^{i^T} e = w, \phi^i \in \Delta^i\} = \{\phi^i \mid \phi^{i^T} e \leq w, \phi^i \in \Delta^i\} \cap \{\phi^i \mid \phi^{i^T} e = w\},$$

which is a closed subset of $\{\phi^i \mid \phi^{i^T} e \leq w, \phi^i \in \Delta^i\}$. Hence 5(b) implies that $\{\phi^i \mid \phi^{i^T} e = w, \phi^i \in \Delta^i\}$ is also compact for any w.

4.2.2 Solution for the model

Recall the model we need to solve is

$$
\begin{array}{llll}
(RP_2) & \max\limits_{\phi} & \min\limits_{r^i \in V^i, y \in U} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) & \max\limits_{\phi^i} \quad \min\limits_{r^{ij} \in V^{ij}, y^i \in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) u(\phi^{i^T} r^{ij}) \\
& & & \text{s.t.} \quad \phi^{i^T} e = \phi^T r^i, \forall i = 1, 2, ..., m \\
& & & \qquad \phi^i \in \Delta^i \\
& \text{s.t.} & \phi^T e = 1 & \\
& & \phi \in \Delta. &
\end{array}
$$

Lemma 4.2

$$
\begin{array}{ll}
\max\limits_{\phi^i} & f(\phi^i) \\
\text{s.t.} & \phi^{i^T} e = w_i \\
& \phi^i \in \Delta^i
\end{array} \tag{13}
$$

and

$$
\begin{array}{ll}
\max\limits_{\phi^i} & f(\phi^i) \\
\text{s.t.} & \phi^{i^T} e \leq w_i \\
& \phi^i \in \Delta^i
\end{array} \tag{14}
$$

have the same optimal values, where

$$
f(\phi^i) = \min_{r^{ij} \in V^{ij}, y^i \in U^i} \sum_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) u(\phi^{i^T} r^{ij}).
$$

Proof. First, assumptions 4(b) and 5(b) ensure that both (13) and (14) are solvable, given the fact that these two problems are not infeasible or unbounded. If $\phi_a^{i^*}$ is optimal to (13), then it must be feasible to (14), which means the optimal value of (13) is smaller than or equals to that of (14). Conversely, if $\phi_b^{i^*}$ is optimal to (14), then we can take $\phi_a^i = (w_i - \phi_b^{i^{*T}} e) e_1 + \phi_b^{i^*}$, which is clearly feasible to (13). According to our assumption,

$$
\begin{array}{ll}
& \phi_a^{i^T} r^{ij} \geq \phi_b^{i^*T} r^{ij}, \forall r^{ij} \in V^{ij} \\
\Longrightarrow & u(\phi_a^{i^T} r^{ij}) \geq u(\phi_b^{i^*T} r^{ij}), \forall r^{ij} \in V^{ij} \\
\Longrightarrow & \min\limits_{r^{ij} \in V^{ij}} u(\phi_a^{i^T} r^{ij}) \geq \min\limits_{r^{ij} \in V^{ij}} u(\phi_b^{i^*T} r^{ij}) \\
\Longrightarrow & \sum\limits_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) \min\limits_{r^{ij} \in V^{ij}} u(\phi_a^{i^T} r^{ij}) \geq \sum\limits_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) \min\limits_{r^{ij} \in V^{ij}} u(\phi_b^{i^*T} r^{ij}), \forall y^i \in U^i \\
\Longrightarrow & \min\limits_{y^i \in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) \min\limits_{r^{ij} \in V^{ij}} u(\phi_a^{i^T} r^{ij}) \geq \min\limits_{y^i \in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}_j^i + y_j^i) \min\limits_{r^{ij} \in V^{ij}} u(\phi_b^{i^*T} r^{ij}) \\
\Longrightarrow & f(\phi_a^i) \geq f(\phi_b^{i^*}).
\end{array}
$$

Hence (13) has an optimal value greater than or equal to that of (14). We conclude (13) and (14) have the same optimal value. □

Therefore, under Assumptions 1-5, (RP_2) has the same solution as

$$\begin{array}{lll} \max\limits_{\phi} & \min\limits_{y\in U} \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) & \max\limits_{\phi^i} \quad \min\limits_{r^{ij}\in V^{ij}, y^i\in U^i} \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j) u({\phi^i}^T r^{ij}) \\ & & \text{s.t.} \quad {\phi^i}^T e \le \phi^T r^i, \forall r^i \in V^i \\ & & \qquad\ \ \phi^i \in \Delta^i \\ \text{s.t.} & \phi^T e = 1 & \\ & \phi \in \Delta. & \end{array}$$

Due to the separability and the minimax theorem, we can transform (RP_2) into

$$\begin{array}{ll} \max\limits_{\phi,\phi^i,u_{ij},w_{ij},t_i,t_0} & t_0 \\ \text{s.t.} & t_0 \le \sum\limits_{i=1}^{m} (\tilde{\pi}_i + y_i) t_i, \forall y \in U \\ & t_i \le \sum\limits_{j=1}^{m} (\tilde{\pi}^i_j + y^i_j) u_{ij}, \forall y^i \in U^i \\ & u_{ij} \le u(w_{ij}) \\ & w_{ij} \le {\phi^i}^T r^{ij}, \forall r^{ij} \in V^{ij} \\ & {\phi^i}^T e \le \phi^T r^i, \forall r^i \in V^i \\ & \phi^i \in \Delta, \phi^T e = 1, \phi \in \Delta^i. \end{array}$$

Setting $\mathbf{t} = (t_1, \cdots, t_m)^T$, we have shown in Section 3 that

$$t_0 \le \sum_{i=1}^{m} (\tilde{\pi}_i + y_i) t_i, \forall y \in U$$

can be written as

$$\begin{pmatrix} \tilde{\pi}^T t - t_0 \\ t \end{pmatrix} \in \mathsf{H}(U)^*.$$

By setting $\mathbf{u_i} = (u_{i1}, \cdots, u_{im})^T$, we can apply similar transformations on other constraints and obtain the following finite convex representation of (RP_2):

$$\begin{array}{ll} \max\limits_{\phi,\phi^i,u_{ij},w_{ij},w_i,t_0} & t_0 \\ \text{s.t.} & \begin{pmatrix} \tilde{\pi}^T t - t_0 \\ t \end{pmatrix} \in \mathsf{H}(U)^*, \begin{pmatrix} {\tilde{\pi}^i}^T \mathbf{u_i} - t_i \\ \mathbf{u_i} \end{pmatrix} \in \mathsf{H}(U^i)^* \\ & u_{ij} \le u(w_{ij}) \\ & \begin{pmatrix} -w_{ij} \\ \phi^i \end{pmatrix} \in \mathsf{H}(V^{ij})^*, \begin{pmatrix} -{\phi^i}^T e \\ \phi \end{pmatrix} \in \mathsf{H}(V^i)^* \\ & \phi^i \in \Delta, \phi^T e = 1, \phi \in \Delta^i. \end{array} \tag{15}$$

4.2.3 General model with ellipsoidal ambiguity sets

Obviously, (RP_2) is far more general than (SRP_2). In practice, however, it is arguable that the ellipsoidal ambiguity sets are general enough. Due to statistical considerations, an investor might choose a different utility function from the negate of semi-variance downsiderisk while keeping ambiguity sets ellipsoidal. Therefore, in this section, we assume the ambiguity sets are ellipsoids as defined in Section 3, that is,

$$\begin{array}{l} U = \{y \in \Re^m \mid y^T e = 0, \|y\| \leq \theta\}, \\ V^i = \{r^i \in \Re^n \mid (r^i - \tilde{r}^i)^T Q^i (r^i - \tilde{r}^i) \leq \rho_i^2\}, \\ U^i = \{y^i \mid {y^i}^T e = 0, \|y^i\| \leq \theta_i\}, \\ V^{ij} = \{r^{ij} \mid (r^{ij} - \tilde{r}^{ij})^T Q^{ij} (r^{ij} - \tilde{r}^{ij}) \leq \rho_{ij}^2\}. \end{array}$$

A useful property to note is that if K_1 and K_2 are two convex cones, then

$$K_1^* \cap K_2^* = (K_1 + K_2)^*,$$

where $K_1 + K_2 = \{x + y \mid x \in K_1, y \in K_2\}$. We have

$$\begin{aligned} \mathsf{H}(U) &= \mathrm{cl} \left\{ \begin{pmatrix} y_0 \\ y \end{pmatrix} \middle| y_0 > 0, \|\tfrac{y}{y_0}\| \leq \theta, \tfrac{y^T e}{y_0} = 0 \right\} \\ &= \left\{ \begin{pmatrix} y_0 \\ y \end{pmatrix} \middle| \begin{pmatrix} \theta & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} y_0 \\ y \end{pmatrix} \in \mathrm{SOC}\ (m+1) \right\} \bigcap \left\{ \begin{pmatrix} y_0 \\ y \end{pmatrix} \middle| (y_0\ \ y^T) \begin{pmatrix} 0 \\ e \end{pmatrix} = 0 \right\} \\ &= \left\{ \begin{pmatrix} \theta & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} y_0 \\ y \end{pmatrix} \middle| \begin{pmatrix} y_0 \\ y \end{pmatrix} \in \mathrm{SOC}\ (m+1) \right\}^* \bigcap \left\{ v \begin{pmatrix} 0 \\ e \end{pmatrix} \middle| v \in \Re \right\}^*. \end{aligned}$$

We thus obtain,

$$\mathsf{H}(U)^* = \left\{ \begin{pmatrix} \theta & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} y_0 \\ y \end{pmatrix} + v \begin{pmatrix} 0 \\ e \end{pmatrix} \middle| \begin{pmatrix} y_0 \\ y \end{pmatrix} \in \mathrm{SOC}\ (m+1), v \in \Re \right\}.$$

Therefore, the condition

$$\begin{pmatrix} \tilde{\pi}^T t - t_0 \\ t \end{pmatrix} \in \mathsf{H}(U)^*$$

is equivalent to: $\exists v \in \Re$ such that

$$\begin{pmatrix} \frac{1}{\theta}(\tilde{\pi}^T t - t_0) \\ t - v \cdot e \end{pmatrix} \in \mathrm{SOC}\ (m+1),$$

or

$$\begin{pmatrix} \frac{1}{\theta}(\tilde{\pi}^T t - t_0) \\ t - e \cdot \frac{t^T e}{m} \end{pmatrix} \in \mathrm{SOC}\ (m+1).$$

For the return ambiguity, note that each positive semidefinite matrix Q^i of rank k ($k \le n$) can be decomposed as $Q^i = C^i C^{i^T}$ for some $n \times k$ matrix C^i of full column rank. So the ambiguity set can be written as

$$V^i = \{r^i \in \Re^n \mid \|C^{i^T}(r^i - \tilde{r}^i)\| \le \rho_i\}.$$

Then,

$$\begin{aligned}\mathsf{H}(V^i) &= \mathrm{cl}\left\{\left(\begin{array}{c} r_0^i \\ r^i \end{array}\right)\middle| r_0^i > 0,\ \|C^{i^T}\left(\frac{\tilde{r}^i}{r_0^i} - \frac{r^i}{r_0^i}\right)\| \le \rho_i\right\} \\ &= \left\{\left(\begin{array}{c} r_0^i \\ r^i \end{array}\right)\middle|\left(\begin{array}{cc} 1 & 0 \\ 0 & C^{i^T} \end{array}\right)\left(\begin{array}{cc} \rho_i & 0 \\ \tilde{r}^i & -I \end{array}\right)\left(\begin{array}{c} r_0^i \\ r^i \end{array}\right) \in \mathrm{SOC}\ (n+1)\right\}\end{aligned}$$

leading to

$$\mathsf{H}(V^i)^* = \left\{\left(\begin{array}{cc} \rho_i & \tilde{r}^{i^T} \\ 0 & -I \end{array}\right)\left(\begin{array}{cc} 1 & 0 \\ 0 & C^i \end{array}\right)\left(\begin{array}{c} r_0^i \\ r^i \end{array}\right)\middle|\left(\begin{array}{c} r_0^i \\ r^i \end{array}\right) \in \mathrm{SOC}\ (n+1)\right\}.$$

Therefore,

$$\left(\begin{array}{c} -\phi^{i^T} e \\ \phi \end{array}\right) \in \mathsf{H}(V^i)^*$$

is equivalent to

$$\left(\begin{array}{cc} 1 & 0 \\ 0 & (C^{i^T} C^i)^{-1} C^{i^T} \end{array}\right)\left(\begin{array}{c} \frac{\phi^T r^i - \phi^{i^T} e}{\rho_i} \\ -\phi \end{array}\right) \in \mathrm{SOC}\ (n+1).$$

In fact, the above results can also be obtained by applying the so-called S-lemma (see [3]).

Obviously, ambiguity sets concerning the second decision period can be treated the same way. Therefore, the robust counterpart we want to solve should be

$$\begin{aligned}\min_{\phi, \phi^i, u_{ij}, t_i, t_0} \quad & t_0 \\ \text{s.t.} \quad & \left(\begin{array}{c} \frac{1}{\theta}(t_0 - \pi^T t) \\ e \cdot \frac{t^T e}{m} - t \end{array}\right) \in \mathrm{SOC}\ (m+1), \left(\begin{array}{c} \frac{1}{\theta_i}(t_i - \pi^{i^T} \mathbf{u_i}) \\ e \cdot \frac{\mathbf{u_i}^T e}{m} - \mathbf{u_i} \end{array}\right) \in \mathrm{SOC}\ (m+1), \forall i = 1, 2, ..., m \\ & u_{ij} \le u(w_{ij}) \\ & \left(\begin{array}{cc} 1 & 0 \\ 0 & (C^{i^T} C^i)^{-1} \end{array}\right)\left(\begin{array}{cc} 1 & 0 \\ 0 & C^{i^T} \end{array}\right)\left(\begin{array}{c} \frac{1}{\rho_i}(\phi^T \tilde{r}^i - \phi^{i^T} e) \\ \phi \end{array}\right) \in \mathrm{SOC}\ (n+1) \\ & \left(\begin{array}{cc} 1 & 0 \\ 0 & (C^{ij^T} C^{ij})^{-1} \end{array}\right)\left(\begin{array}{cc} 1 & 0 \\ 0 & C^{ij^T} \end{array}\right)\left(\begin{array}{c} \frac{1}{\rho_{ij}}(\phi^{i^T} \tilde{r}^{ij} - w_{ij}) \\ \phi^i \end{array}\right) \in \mathrm{SOC}\ (n+1) \\ & \phi^i \ge 0, \phi^T e = 1, \phi \ge 0,\end{aligned}$$

where C^{ij} is the full rank matrix satisfying $C^{ij}C^{ij^T} = Q^{ij}$ for $i, j = 1, 2, ..., m$. After replacing t_0 with $-t_0$, t_i with $-t_i$ and u_{ij} with $-d_{ij}$, our model becomes

$$\begin{array}{ll}\min\limits_{\phi,\phi^i,d_{ij},t_i,t_0} & t_0 \\ \text{s.t.} & \begin{pmatrix} \frac{1}{\theta}(t_0 - \pi^T t) \\ e \cdot \frac{t^T e}{m} - t \end{pmatrix} \in \text{SOC } (m+1), \begin{pmatrix} \frac{1}{\theta_i}(t_i - \pi^{i^T}\mathbf{d_i}) \\ e \cdot \frac{\mathbf{d_i}^T e}{m} - \mathbf{d_i} \end{pmatrix} \in \text{SOC } (m+1), \forall i = 1, 2, ..., m \\ & -d_{ij} \leq u(w_{ij}) \\ & \begin{pmatrix} 1 & 0 \\ 0 & (C^{i^T}C^i)^{-1} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & C^{i^T} \end{pmatrix}\begin{pmatrix} \frac{1}{\rho_i}(\phi^T\tilde{r}^i - \phi^{i^T}e) \\ \phi \end{pmatrix} \in \text{SOC } (n+1) \\ & \begin{pmatrix} 1 & 0 \\ 0 & (C^{ij^T}C^{ij})^{-1} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & C^{ij^T} \end{pmatrix}\begin{pmatrix} \frac{1}{\rho_{ij}}(\phi^{i^T}\tilde{r}^{ij} - w_{ij}) \\ \phi^i \end{pmatrix} \in \text{SOC } (n+1) \\ & \phi^i \geq 0, \phi^T e = 1, \phi \geq 0. \end{array} \tag{16}$$

To simplify, we just take $Q^i = Q^{ij} = \begin{pmatrix} 0 & 0 \\ 0 & I_{n-1} \end{pmatrix}$, and then $C^i = C^{ij} = \begin{pmatrix} 0 \\ I_{n-1} \end{pmatrix}$ for all i, j. In that case, (16) becomes

$$\begin{array}{ll}\min\limits_{\phi,\phi^i,u_{ij},t_i,t_0} & t_0 \\ \text{s.t.} & \begin{pmatrix} t_0 - \pi^T t \\ \theta(e \cdot \frac{t^T e}{m} - t) \end{pmatrix} \in \text{SOC } (m+1), \begin{pmatrix} t_i - \pi^{i^T}\mathbf{u_i} \\ \theta_i(e \cdot \frac{\mathbf{d_i}^T e}{m} - \mathbf{d_i}) \end{pmatrix} \in \text{SOC } (m+1), \forall i = 1, 2, ..., m \\ & -d_{ij} \leq u(w_{ij}) \\ & \begin{pmatrix} \phi^T\tilde{r}^i - \phi^{i^T}e \\ 0 \\ \rho_i\phi_2 \\ \vdots \\ \rho_i\phi_n \end{pmatrix} \in \text{SOC } (n+1), \begin{pmatrix} \phi^{i^T}\tilde{r}^{ij} - w_{ij} \\ 0 \\ \rho_{ij}\phi^i_2 \\ \vdots \\ \rho_{ij}\phi^i_n \end{pmatrix} \in \text{SOC } (n+1) \\ & \phi^i \geq 0, \phi^T e = 1, \phi \geq 0. \end{array} \tag{17}$$

In our numerical experiments, a particular penalty function method is used to solve the above model. We shall discuss more details in the next section.

5 Numerical results and concluding remarks

In this section we report the results of our preliminary computational experiments under the robust portfolio selection framework proposed in the previous chapters. Although the model can be in principle applied to any multi-stage settings, our tests are done only on the two-stage problems as specified in Section 4.

5.1 Scenario tree generation

For implementation, the first issue is to generate a scenario tree which to be used in the deterministic problem (P_2). We choose to adopt the "clustering of parallel simulations method" as introduced in Gulpinar et al ([8], 2004). The main idea is to partition the simulated scenarios in random clusters and select one scenario in each cluster as a representative, which is known as "centroid". The steps of their algorithm are quoted as follows:

Step 1 (Initialization): Create a root node, with N scenarios. Initialize all the scenarios (including the centroid) with the desired starting point. Form a job queue consisting of the root node.

Step 2 (Simulation): Remove a node from the job queue. Simulate one time period of growth in each scenario.

Step 3 (Randomized seeds): Randomly choose a number of distinct scenarios around which to cluster the rest: one per desired branch on the scenario tree.

Step 4 (Clustering): Group each scenario with the seed point to which it is the closest. If the resulting clustering is unacceptable, return to *Step 3*.

Step 5 (Centroid selection): For each cluster, find the scenario which is the closest to its center, and designate it as the centroid.

Step 6 (Queueing): Create a child scenario tree node for each cluster (with probability proportional to the number of scenarios in the cluster), and install its scenarios and centroid. If the child nodes are not leaves, append to the job queue. If the queue is non-empty, return to *Step 2*. Otherwise, terminate the algorithm.

In our actual implementations, we simply consider perfect trees, in which every parent node has exactly m children. We generate $N = 10m^2$ simulations and then partition them into m^2 clusters. This means each cluster will contain 10 simulations on average. Each simulation is generated from a vector autoregressive (VAR) model:

$$h_t = c + \Omega h_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, \Sigma), \qquad t = 1, ..., T.$$

Note h_t so generated are continuous rates. After the tree is generated, we should use $\tilde{r}_t = e^{h_t}$ on each node to obtain the vectors of the discrete returns.

Using the VAR model to generate the scenario tree is not a new approach. Such experiments have been done in Kouwenberg (2001) for asset liability management. Kouwenberg also used some other scenario tree generation methods in the paper. In our numerical tests, historical financial data will be used to fit the VAR model.

5.2 Numerical results for an instance of (RSP_2)

Suppose that we want to choose a portfolio among four indices: Heng Seng Index, Dow Jones index, London index and Nikkei. The decision horizon is divided into two periods, and the length of each period is one month. The target return is assumed to be 0.3% for these two months in total, i.e R=1.003. We use the monthly price from Jan. 2001 to Dec. 2004[2] as historical data to get a least square estimate for the VAR model:

$$h_t = c + \Omega h_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, \Sigma), \qquad t = 1, ..., T$$

The fitted parameters are summarized below.

$$\hat{c} = \begin{pmatrix} 0.0026 \\ 0.0029 \\ -0.0036 \\ 0.0004 \end{pmatrix}, \hat{\Omega} = \begin{pmatrix} 0.1548 & -0.2725 & 0.3374 & 0.2171 \\ -0.0156 & -0.2432 & 0.2633 & 0.1672 \\ 0.0036 & -0.0703 & 0.0178 & 0.1549 \\ 0.0769 & -0.1888 & 0.3729 & 0.0852 \end{pmatrix},$$

$$\hat{\Sigma} = \begin{pmatrix} 0.0029 & 0.0019 & 0.0015 & 0.0013 \\ 0.0019 & 0.0022 & 0.0018 & 0.0011 \\ 0.0015 & 0.0018 & 0.0019 & 0.0010 \\ 0.0013 & 0.0011 & 0.0010 & 0.0025 \end{pmatrix}.$$

Note all the eigenvalues of $\hat{\Omega}$ lie in the unit ball so that the fitted VAR is stationary. A perfect two-stage scenario tree with m children on each internal node can be generated from this specific VAR model by the clustering method as described in the previous section. We randomly generate a tree of degree $m = 10$ and a tree of degree $m = 20$ respectively. Based on each estimated tree, we solve (SP_2) and its robust counterpart and obtain the optimal initial portfolios ϕ_{SP_2} and ϕ_{RSP_2} respectively. To compare the two portfolios, we generate 30 realized trees in which return vectors r^i are chosen by the following procedure:

Step 1. Choose l from a uniform distribution on $[0, \rho_i]$;

Step 2. Choose $\alpha_1, \alpha_2, ..., \alpha_{n-1}$ from independent uniform distributions on $[0, 2\pi]$;

[2]Source: www.yahoo.com

Step 3. Set

$$\begin{cases} \Delta r^i(1) & = & l \sin \alpha_1, \\ \Delta r^i(2) & = & l \cos \alpha_1 \sin \alpha_2, \\ & \vdots & \\ \Delta r^i(n-1) & = & l \cos \alpha_1 \cdots \cos \alpha_{n-1} \sin \alpha_n \\ \Delta r^i(n) & = & l \cos \alpha_1 \cdots \cos \alpha_n. \end{cases}$$

Step 4. Set $r^i = \text{perm}(\Delta r^i) + \tilde{r}^i$, where $\text{perm}(\cdot)$ returns a random permutation of a vector.

To simulate the probability vectors is a little complicated because we have to ensure $\pi^T e = 1$ and ${\pi^i}^T e = 1$. The following procedure is adopted to generate such π:

Step 1. Simulate a vector $z \in \Re^{m-1}$ satisfying $\|z\| \leq \sqrt{\frac{m-2+a^2}{2}}\theta$, where $a = 2 + \sqrt{m}$;
Step 2. Set $y = \begin{pmatrix} A^{-1}z \\ -e^T A^{-1} z \end{pmatrix}$,
where $A = \begin{pmatrix} a & 1 & \cdots & 1 \\ 1 & a & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & a \end{pmatrix} \in \Re^{(m-1)\times(m-1)}$;
Step 3. Set $y = \text{perm}(y)$;
Step 4. Set $\pi = \tilde{\pi} + y$.

The other π^is and r^{ij}s are generated similarly. Since we are allowed to adjust our portfolio at the end of period 1 when the stage-one return r^is are realized, the recourse problem and its robust counterpart need to be solved. That is, we should obtain the appropriate stage-two portfolios $\phi^i_{SP_2}$ and $\phi^i_{RSP_2}$ by solving

$$\begin{array}{ll} \min\limits_{\phi^i} & \sum\limits_{j=1}^m \tilde{\pi}^i_j (R - {\phi^i}^T \tilde{r}^{ij})^2_+ \\ \text{s.t.} & {\phi^i}^T e = \phi^T_{SP_2} r^i, \forall i = 1, 2, ..., m \\ & \phi^i \geq 0 \end{array}$$

and

$$\begin{array}{ll} \min\limits_{\phi^i} & \max\limits_{r^{ij} \in V^{ij}, y^i \in U^i} \sum\limits_{j=1}^m (\tilde{\pi}^i_j + y^i_j)(R - {\phi^i}^T r^{ij})^2_+ \\ \text{s.t.} & {\phi^i}^T e = \phi^T_{RSP_2} r^i, \forall i = 1, 2, ..., m \\ & \phi^i \geq 0 \end{array}$$

respectively. Note the recourse problem and its robust counterpart can be viewed as a single-stage portfolio selection problem and its robust counterpart respectively and therefore can be written respectively as the following SOCPs:

$$\begin{array}{cl}\min\limits_{\phi^i,\tau_j,t_j,t_0} & t_0 \\ \text{s.t.} & t_0 - \tilde{\pi}^{i^T} t \geq 0 \\ & \begin{pmatrix} t_j+1 \\ t_j-1 \\ 2\tau_j \end{pmatrix} \in \text{SOC } (3) \\ & \tau_j - R + \phi^{i^T} \tilde{r}^{ij} \geq 0 \\ & \tau_j \geq 0, \forall j = 1, ..., m \\ & \phi^{i^T} e = \phi^T_{SP_2} r^i \\ & \phi^i \geq 0 \end{array}$$

and

$$\begin{array}{cl}\min\limits_{\phi^i,\tau_j,t_j,t_0} & t_0 \\ \text{s.t.} & \begin{pmatrix} t_0 - \tilde{\pi}^{i^T} t \\ \theta_i(e \cdot \frac{t^T e}{m} - t) \end{pmatrix} \in \text{SOC } (m+1) \\ & \begin{pmatrix} t_j+1 \\ t_j-1 \\ 2\tau_j \end{pmatrix} \in \text{SOC } (3), \begin{pmatrix} \tau_j - R + \phi^{i^T} \tilde{r}^{ij} \\ \rho_{ij}\phi^i \end{pmatrix} \in \text{SOC } (n+1) \\ & \tau_j \geq 0, \forall j = 1, ..., m \\ & \phi^{i^T} e = \phi^T_{RSP_2} r^i \\ & \phi^i \geq 0. \end{array}$$

For each of the 30 simulated trees, we simulate 500 scenario paths. With the above stage-one and stage-two portfolios, we compute the downside risks for each path on each simulated tree. The results are summarized in Tables 1 to 4. Then, for each parameter setting, we calculate the mean, standard deviation, minimum and maximum of the mean of 500 disutility values. The results are then summarized in Table 5.

As we may observe from Table 5, the standard deviation of the mean of the downside risk will be significantly reduced if we choose portfolio ϕ_{RSP} instead of ϕ_{SP} without sacrificing the mean return. Even though the expected downside risk has increased, the increment is relatively small and is well compensated by the reduction in the standard deviation (volatility) of the objective value due to the model ambiguity. For example, when $m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$, using a robust optimal portfolio will only cause a 7% increase in the mean of the expected downside risk, while the standard deviation is decreased by over 40%. At the same, the worst objective value (the mean of downside risk) for each of the 30 simulated trees also decrease. This is consistent with our theoretical results. When $m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$, for instance, the objective value in the worst case decreases by a factor of 15%. Finally, we observe that for the same uncertain scenario tree, increasing the size of the ambiguity sets in the robust optimization model will lead to the reduction in the standard deviation of the objective values, and the objective values in the worst cases at the same time.

Although we plan to do more numerical experiments to further test our robust multi-stage portfolio investment strategies (this will be the topic of our future research), the results we obtained so far have indicated that the robust considerations are important in that the investment portfolios indeed become less sensitive to the parameters in the model (which is a necessary feature for any investment strategy to be successful), at a reasonable cost of slightly lowering average return rates. Finally, we draw the following general conclusion from our investigations. In the presence of parameter ambiguities, the minimax style robust immunization helps to stabilize the undesired solution sensitivity in the scenario-tree based multi-stage stochastic optimization models, where the variability of the solutions is clearly reduced. The key to success, it appears, is the ability to formulate the computational model in a tractable form. Fortunately, this modelling power of ours has increased over time, thanks to the availability of novel optimization tools that have been recently developed in the field; such tools include Second Order Cone Programming.

References

[1] A. Ben-Tal and A. Nemirovski, *Robust convex optimization*, Mathematics of Operations Research, 23 (1998), pp. 769 – 805.

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

[3] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS-SIAM Series on Optimization, 2001.

[4] M. Breton and S. El Hachem, *Algorithms for the solution of stochastic dynamic minimax problems*, Computational Optimization and Applications, 4 (1995), pp. 317 – 345.

[5] J. Brinkhuis and S. Zhang, *A D-induced duality and its applications*, Technical Report SEEM 2002-01, Department of System Engineering and Engineering Management, The Chinese University of Hong Kong.

[6] J. Dupačová, *The minimax approach to stochastic programming and an illustrative application*, Stochastics, 20 (1987), pp. 73 – 88.

[7] D. Goldfarb and G. Iyengar, *Robust portfolio selection problems*, Mathematics of Operations Research, 28 (2003), pp. 1 – 38.

[8] N. Gulpinar, B. Rustem and R. Settergren, *Simulation and optimization approaches to scenario tree generation*, Journal of Economic Dynamics & Control, 28 (2004), pp. 1291 – 1315.

[9] R. Kouwenberg, *Scenario generation and stochastic programming models for asset liability management*, European Journal of Operational Research, 134 (2001), pp. 279 – 292.

[10] Z.Q. Luo, J.F. Sturm and S. Zhang, *Multivariate nonnegative quadratic mappings*, SIAM Journal on Optimization, 14 (2004), pp. 1140 – 1162.

[11] B. Rustem and M. Howe, *Algorithms for Worst-Case Design and Applications to Risk Management*, Princeton University Press, Princeton, New Jersey, 2002.

[12] S. Simons, *Minimax theorems and their proofs*. In D.Z. Du and P.M. Pardalos eds., *Minimax and Applications*, Kluwer Academic Publishers, Dordrecht, 1995, pp. 1 – 23.

[13] J.F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11-12 (1999), pp. 625-653. Special issue on Interior Point Methods (CD supplement with software).

[14] J.F. Sturm, *Theory and Algorithms for Semidefinite Programming*, in Frenk *et al.* eds., High Performance Optimization, Kluwer Academic Publishers, 1999.

[15] J.F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Mathematics of Operations Research, 28 (2003), pp. 246 – 267.

Simulated Tree No.	mean(ϕ_{SP})	mean(ϕ_{RSP})	std(ϕ_{SP})	std(ϕ_{RSP})
1	0.0151	0.0150	0.0166	0.0169
2	0.0184	0.0183	0.0179	0.0184
3	0.0167	0.0167	0.0175	0.0178
4	0.0185	0.0175	0.0184	0.0182
5	0.0164	0.0162	0.0164	0.0167
6	0.0164	0.0161	0.0171	0.0173
7	0.0171	0.0169	0.0184	0.0186
8	0.0159	0.0159	0.0169	0.0176
9	0.0156	0.0153	0.0164	0.0165
10	0.0174	0.0172	0.0177	0.0179
11	0.0179	0.0176	0.0182	0.0182
12	0.0164	0.0166	0.0172	0.0177
13	0.0178	0.0174	0.0179	0.0182
14	0.0157	0.0156	0.0176	0.0177
15	0.0161	0.0159	0.0164	0.0167
16	0.0154	0.0153	0.0162	0.0164
17	0.0166	0.0165	0.0179	0.0180
18	0.0180	0.0182	0.0181	0.0186
19	0.0155	0.0160	0.0158	0.0164
20	0.0162	0.0158	0.0162	0.0164
21	0.0168	0.0167	0.0174	0.0176
22	0.0170	0.0169	0.0174	0.0177
23	0.0163	0.0157	0.0176	0.0176
24	0.0161	0.0157	0.0168	0.0171
25	0.0164	0.0160	0.0174	0.0175
26	0.0157	0.0160	0.0162	0.0166
27	0.0165	0.0171	0.0165	0.0170
28	0.0169	0.0162	0.0175	0.0174
29	0.0161	0.0165	0.0173	0.0179
30	0.0173	0.0173	0.0176	0.0180

Table 1: The comparison of optimal expected utility function values for (SP_2) and (RSP_2) when $m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$.

Simulation No.	mean(ϕ_{SP})	mean(ϕ_{RSP})	std(ϕ_{SP})	std(ϕ_{RSP})
1	0.023163	0.020068	0.021263	0.019548
2	0.044821	0.029962	0.025681	0.022144
3	0.035555	0.014659	0.025430	0.016782
4	0.009818	0.015120	0.013720	0.017967
5	0.020192	0.019212	0.017551	0.016757
6	0.007236	0.011178	0.012395	0.015890
7	0.017326	0.020581	0.017967	0.017165
8	0.019015	0.023077	0.017436	0.018262
9	0.018063	0.017905	0.018367	0.017897
10	0.024886	0.019590	0.021360	0.017593
11	0.013507	0.017381	0.015518	0.017839
12	0.029619	0.026344	0.022018	0.022391
13	0.016177	0.017482	0.017998	0.019241
14	0.014998	0.022545	0.012095	0.017163
15	0.025942	0.023584	0.029900	0.023360
16	0.037115	0.025366	0.027865	0.019414
17	0.007683	0.018293	0.007484	0.018316
18	0.016520	0.018605	0.017766	0.018539
19	0.015087	0.016467	0.016813	0.018346
20	0.011286	0.018490	0.014919	0.018240
21	0.020323	0.021368	0.018900	0.019210
22	0.012855	0.016592	0.015141	0.016435
23	0.015040	0.016968	0.021476	0.018911
24	0.018698	0.019387	0.020483	0.020112
25	0.026605	0.012250	0.020975	0.014916
26	0.015211	0.019168	0.016939	0.017204
27	0.017973	0.017714	0.017324	0.015727
28	0.022325	0.018573	0.012776	0.014783
29	0.019459	0.019266	0.026209	0.019811
30	0.013717	0.015176	0.018941	0.017066

Table 2: The comparison of optimal expected utility function values for (SP_2) and (RSP_2) when $m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$.

Simulation No.	mean(ϕ_{SP})	mean(ϕ_{RSP})	std(ϕ_{SP})	std(ϕ_{RSP})
1	0.039628	0.039809	0.023097	0.023488
2	0.037118	0.036793	0.021986	0.022116
3	0.036690	0.036461	0.021469	0.021797
4	0.039884	0.039497	0.022867	0.023090
5	0.039111	0.039200	0.021895	0.022279
6	0.038316	0.038526	0.023108	0.023433
7	0.036741	0.036993	0.021604	0.021838
8	0.038177	0.038237	0.021965	0.022210
9	0.038651	0.038957	0.023867	0.024193
10	0.034983	0.035368	0.020252	0.020658
11	0.035605	0.035909	0.021934	0.022192
12	0.038245	0.038696	0.022896	0.023142
13	0.038517	0.038092	0.021979	0.022197
14	0.036701	0.036769	0.021526	0.021871
15	0.037487	0.037608	0.021679	0.022021
16	0.039209	0.039359	0.022289	0.022775
17	0.037823	0.037899	0.022457	0.022917
18	0.040010	0.040258	0.022554	0.022849
19	0.038135	0.038392	0.021934	0.022075
20	0.037735	0.037789	0.021149	0.021553
21	0.040037	0.039216	0.023104	0.023249
22	0.038217	0.037867	0.021042	0.021400
23	0.038830	0.039220	0.024004	0.024446
24	0.037897	0.038142	0.022188	0.022521
25	0.037257	0.037431	0.021945	0.022323
26	0.039026	0.038574	0.022346	0.022836
27	0.037971	0.038250	0.020771	0.021314
28	0.037358	0.037433	0.021890	0.022544
29	0.039180	0.039579	0.023111	0.023347
30	0.037915	0.037970	0.022341	0.022711

Table 3: The comparison of optimal expected utility function values for (SP_2) and (RSP_2) when $m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$.

Simulation No.	mean(ϕ_{SP})	mean(ϕ_{RSP})	std(ϕ_{SP})	std(ϕ_{RSP})
1	0.042777	0.046607	0.026811	0.028027
2	0.037954	0.042539	0.023226	0.024736
3	0.056140	0.041825	0.030826	0.023360
4	0.032971	0.037870	0.019647	0.022278
5	0.040384	0.051911	0.023510	0.026385
6	0.063347	0.044975	0.029049	0.023734
7	0.041037	0.042239	0.022407	0.022198
8	0.044176	0.042903	0.034973	0.025256
9	0.032981	0.042085	0.020581	0.025931
10	0.034207	0.044676	0.023391	0.024808
11	0.044625	0.042605	0.025458	0.024451
12	0.056850	0.053531	0.029651	0.031299
13	0.055637	0.046887	0.027770	0.023176
14	0.038220	0.045294	0.026780	0.027837
15	0.036118	0.048949	0.021267	0.023266
16	0.048894	0.050769	0.024468	0.025263
17	0.031945	0.036074	0.022031	0.021659
18	0.024528	0.031694	0.018670	0.020064
19	0.043542	0.043192	0.025795	0.026158
20	0.042073	0.042038	0.024343	0.024527
21	0.029920	0.035966	0.020121	0.021963
22	0.039434	0.044167	0.021728	0.022732
23	0.039135	0.041967	0.022928	0.022829
24	0.039541	0.045644	0.025274	0.024465
25	0.044885	0.045840	0.025910	0.023839
26	0.042935	0.053967	0.023734	0.027428
27	0.027826	0.038244	0.018155	0.019736
28	0.033608	0.041651	0.021098	0.023825
29	0.044948	0.043498	0.020028	0.020387
30	0.037101	0.044094	0.017852	0.026881

Table 4: The comparison of optimal expected utility function values for (SP_2) and (RSP_2) when $m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$.

Parameter settings	mean(mean(ϕ_{SP}))	mean(mean(ϕ_{RSP}))
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.016594	0.016472
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.019674	0.019079
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.038082	0.038143
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.040925	0.043790
Parameter settings	std(mean(ϕ_{SP}))	std(mean(ϕ_{RSP}))
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.000896	0.000838
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.008564	0.003961
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.001217	0.001168
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.008844	0.005006
Parameter settings	min(mean(ϕ_{SP}))	min(mean(ϕ_{RSP}))
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.015113	0.015049
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.007236	0.011178
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.034983	0.035368
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.024528	0.031694
Parameter settings	max(mean(ϕ_{SP}))	max(mean(ϕ_{RSP}))
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.018534	0.018310
$m = 10, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.044821	0.029962
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.01$	0.040037	0.040258
$m = 20, \theta = \theta_i = \rho_i = \rho_{ij} = 0.1$	0.063347	0.053967

Table 5: The comparison of the mean of 500 simulated disutility function values for (SP_2) and (RSP_2) under different parameter settings.