A New Simulation-Based Approach for Multi-Period Portfolio Optimization Problems

Guidance

Professor Masao FUKUSHIMA
Assistant Professor YAMASHITA

HAIGUANG HU

2001 Graduate Course

in

Department of Applied Mathematics and Physics

Graduated School of Informatics

Kyoto University

February 2003

Abstract

As another alternative approximation method for multi-period dynamic portfolio selection problems, a new model, called the hybrid simulation tree model, has been proposed recently. In that model, the underlying uncertainty is revealed as a simulation tree and, similar to the scenario tree model, different conditional decisions are allowed to be made for different bundles (or groups) of sample paths which exhibit similar performance characteristics. By adopting a fixed-quantity policy, it was formulated as a large-scale linear programming problem, in which the compromise between describing the precision of the underlying uncertainty and making conditional decisions could be well handled within a single framework. Such a model is much more general than the conventional scenario tree model and simulation path model, and is expected to be a promising method with various applications.

We extend this approach and combine it with two new ideas: (i) In order to improve the performance, we use the Conditional Value-at-Risk as our risk measure, which is a very important concept in the modern risk management field. (ii) For consideration of preserving the stability of optimal solutions and hedging a shortfall risk, we incorporate two different types of risk chance constraints in our model. Then the problem is formulated as a Second-Order Cone Programming (SOCP) problem, which can be solved efficiently by using some sophisticated mathematical programming software packages. Numerical experiments are also carried out to demonstrate the efficacy of the proposed model in hedging against risk.

Contents

1 Introduction

In this paper, we will consider a typical portfolio management problem over a long period of time, faced by both individual and institutional investors. Such problems are crucial for all business firms and many other industrial, public, academic and governmental agencies. Starting with a given portfolio, a portfolio manager has to assess the prevailing market information, such as asset prices, interest rates, transaction costs, as well as the existing portfolio composition at each time point along the planning horizon. Such information, in conjunction with predictions of possible future interest rates, asset prices, regulation changes and so forth, is incorporated into an investment decision, resulting in a sequence of selling and buying actions. We call it an observation-reaction process. The goal of a portfolio manager is to determine a sequence of portfolio decisions that maximizes his/her expected utility function or, equivalently, to minimize the risk exposure under various constraints such as budget constraints, investment bound constraints, and regulation restrictions.

Such decision making problems over a long period of time, inherently involving uncertainty, are usually transformed into stochastic programming problems, whose range of applications is as broad as any in optimization. See for instance [6, 17, 18, 7, 30, 15, 9] and the references therein.

The overwhelming difficulty in stochastic programming problems stems from the evaluation of random functions and expectations of random parameters. Since the size as well as the complexity of a stochastic programming problem may grow rapidly as additional uncertainties are modelled. Many of algorithm developments in this area have mainly concentrated on methods to exploit stochastic program structure. For example, Dantzig and Mandansky [8] introduced Dantzig-Wolfe decomposition as a possible solution method, while Van Slyke and Wets [23] developed a form of Benders decomposition [3] for stochastic programmming. Beyond algorithms, much of stochastic programming research has considered various theoretical properties, such as convexity, continuity, and stability properties, etc. See Wets [28] for a general overview of these results.

In fact, without some sort of approximation methods for realistic problems, it is difficult to obtain optimal solutions with closed form using stochastic programming. Thus various approximation schemes, both deterministic and stochastic ones, have been designed. The notion of scenario is typically employed for modelling random parameters in the multi-period stochastic programming (MSP) models. Then the input information of the underlying uncertainty can be represented in the form of a scenario tree or event tree. Each path through the tree from the root to one of its leaves corresponds to one scenario, i.e., to a particular sequence of realizations of the underlying uncertainty. This model is based on the expansion of decision space, taking into account the conditional nature of a scenario tree. Conditional decisions are made at each node of this tree. The goal is to get a numerically tractable optimization problem, or a sequence of such problems, whose solution would be acceptable as an approximate solution of the true underlying problem.

In most cases, given the scenarios and their occurrence probabilities, the result-

ing problem is nothing else but a large-scale linear program of special structure. The optimal value and the optimal solution depend on the choice of scenarios. Modification of scenarios will influence in general the whole structure of the problem. Clearly, the number of scenarios is a crucial factor from the point of view of the problem's size and its numerical tractability.

The primary aim of scenario generation is to represent the underlying uncertainties in a reasonable way. However, this is an ambitious task in which compromise has to be made between describing the precision of the underlying uncertainty and limiting the size of the approximate problem. If a representative set of scenarios is constructed so as to cover the set of possibilities to a sufficient degree, the numbers of decision variables and constraints appearing in the problem may grow exponentially. Especially if the number of asset categories available for portfolio selection is large, it becomes the fatal weakness of this kind of method.

In this paper, inspired by Hibiki [12], we consider an alternative approach for portfolio optimization problems. For a broader decision space than the traditional simulation path-based approach, we allow different decisions to be made for different bundles of sample paths, where sample paths, exhibiting similar performance characteristics, are bundled together for some standard measure. Sample paths are bundled to avoid violation of non-anticipativity [1] and, at the same time, to dramatically limit the number of decision variables. The dimension of the problem thus increases linearly with respect to the number of bundles and the number of time periods. This is an improvement over the exponential growth of problem size in the stochastic programming approach.

The idea of using decision variables independently of scenarios is not new. Financial models based on string (linear) scenario trees instead of event trees have been suggested by Hiller and Eckstein [13] and Zenios [29].

The main contribution of this paper is that we extend this approach with two new ideas: (1) To ensure the stability and robustness of the optimal solution, two different types of risk chance constraints are integrated into our model without destroying the convexity, which means any local optimal solution will be a global optimal one. (2) We combine our model with a new risk management technique, called Conditional Value-at-Risk (CVaR), see Rockafellar and Uryasev [21]. Optimization is then performed directly using a set of simulated sample paths, thereby eliminating the need to construct a scenario tree, which often leads to significant methodological and computational difficulties.

Although, as a current standard in finance industry, Value-at-Risk (VaR) is a popular measure of risk, because of its non-convexity and some other undesirable features, we use Conditional Value-at-Risk (CVaR) in our model as the measure of risk instead. Fundamental properties of CVaR are derived from the loss distribution in finance that can involve discreteness. Such a distribution is of particular importance in realistic applications because of the prevalence of models based on scenarios and finite samplings. CVaR is able to quantify dangers beyond VaR, and moreover

[1] *non-anticipativity*: In each stage, decisions must be made without knowledge of the realizations of random variables in future stages.

it is convex and coherent. (The coherency of CVaR was first proved by Pflug [20]; also see Acerbi and Tasche [1] and Rockefeller and Uryasev [21]). It provides optimization shortcuts which, through linear programming techniques, make practical many large-scale calculations that could otherwise be out of reach. We use the fact that incorporating CVaR constraints does not destroy the convexity of our model, an important feature as a risk measure in optimization settings.

In the framework of our model, clustering analysis plays an important role to classify simulated sample paths into several bundles at each decision making point along the planning horizon so as to generate a decision tree, which we call it a simulation tree in order to distinguish it from a well-known scenario tree. Clustering is a hierarchical agglomeration method for identifying groups of samples in a multivariate data set and remains the most popular classification technique today. However, an important issue is that the results may change drastically if one selects different joining rules. Choice of distance measure can also have an effect, though usually not as drastic as the joining rules. Thus, we have to consider the stability or the robustness of the optimal solution and the optimal value obtained from the model based on the clustering analysis method: The procedure should be robust in the sense that small perturbations of the inputs, *i.e.*, of the chosen bundles, of the sample paths and of the number of simulated paths, should impair the outcome only slightly so that the obtained results remain close to the unperturbed ones and that somewhat large perturbations do not cause a catastrophe. For this propose, we introduce two types of risk chance constraints into our model, which can be converted into Second-Order Cone (SOC) constraints. For the consideration of stability, our results show that it is a worthwhile job. Moreover, through empirical experiments, the optimal solutions obtained from our model show a convergence behavior for a specific joining rule when increasing the number of simulated paths, which indicates that we can control the size of the optimization problem by taking advantage of such characteristics.

The optimization is conducted by formulating the model as a Second-Order Cone Programming (SOCP) problem, which is a special type of convex optimization problem. For the last decade, SOCP has been under intensive study and many important problems like LP, QP, and SDP can be formulated as SOCPs. By the fact that SOCP problems can be solved in polynomial time by interior point methods, many efficient software packages have been well developed. In this paper, we carried out our numerical experiments by using SeDuMi [24], a popular interior point solver for convex minimization problems on a self-dual cone.

The rest of this paper is organized as follows. In the next two sections, we illustrate some modelling aspects of our proposed hybrid simulation tree model and introduce their mathematical formulations. In Section 4, we discuss two possible risk chance constraints which can be converted into SOCs. Section 5 is devoted to the Conditional Value-at-Risk (CVaR), which we integrate in our model. Numerical experiments are carried out in Section 6. Finally, in Section 7, we give a brief conclusion of this paper.

2 Model Description

So far, various approximation schemes have been proposed to describe the dynamics and underlying uncertainties in stochastic programming problems, modelled via explicitly spelling out random parameters. The Monte Carlo approach has proved to be a valuable and flexible computational tool in the modern financial industry. It is easy to get a set of simulation paths representing possible future outcomes of the underlying uncertainty, provided its process is explicitly expressed by a stochastic differential equation or a time series model. Technical aspects of Monte Carlo approach are beyond the scope of this paper, see [14, 5] for references.

The most typical decision rule is the fixed-proportion strategy. Since the products of decision variables emerge in the formulation, this rule leads to a non-convex optimization problem, which indicates a global optimal solution can hardly be obtained. Hence, such a method is not recommended for realistic problems.

An alternative way is to convert a set of simulated sample paths into a scenario tree so that conditional decisions can be made with respect to each node of the scenario tree, without violating the non-anticipativity condition. Since such conversion lets us use the well-developed stochastic optimization methodology to find an optimal solution, this approach is widely accepted and many successful applications have been reported. However, for a multi-period model involving multiple asset categories, converting sample paths into a scenario tree may lead to significant methodological and computational difficulties. The overwhelming problem is that the size of the problem will grow exponentially as the number of child nodes and/or periods increases. On the other hand, for the propose of limiting the size of the problem, one has to reduce the number of scenarios, which in turn severely limits the precision of describing the future possible outcomes of the underlying uncertainties.

With a set of simulated paths, ideally one would like to make different decisions for every path at every time $t = 1, \ldots, T-1$, but this would lead to undesired violation of non-anticipativity condition in the model. This is caused by the fact that, once we start following a specific path, we have full knowledge of the future until time T. The simplest way to avoid anticipativity is to make one single decision at each time t for all paths, that is, to use a path-independent strategy, which means that the decision variables are independent of all path realizations at a given time. This is the basis of current sample path-based approaches.

Even though a linear programming model can be formulated by using the fixed-quantity rule instead of the fixed-proportion rule and can be solved efficiently by any mathematical programming software, such an approach is also inadequate for the multi-period portfolio problem since it lacks flexibility in decision making and cannot reflect the conditional decision making process properly in practice. Some other sample path based approaches have also been considered, *i.e.*, a simulation tree approach by Hibiki [11] and a Grid-Net Scheme [2]. The main idea is to find a proper

[2] At time t, sample paths are grouped in several bundles. Paths that are in two different groups at time t can pass through the same group at either earlier or later time (or both). Also, paths that pass through one node at time t do not necessarily pass through the same node at any other

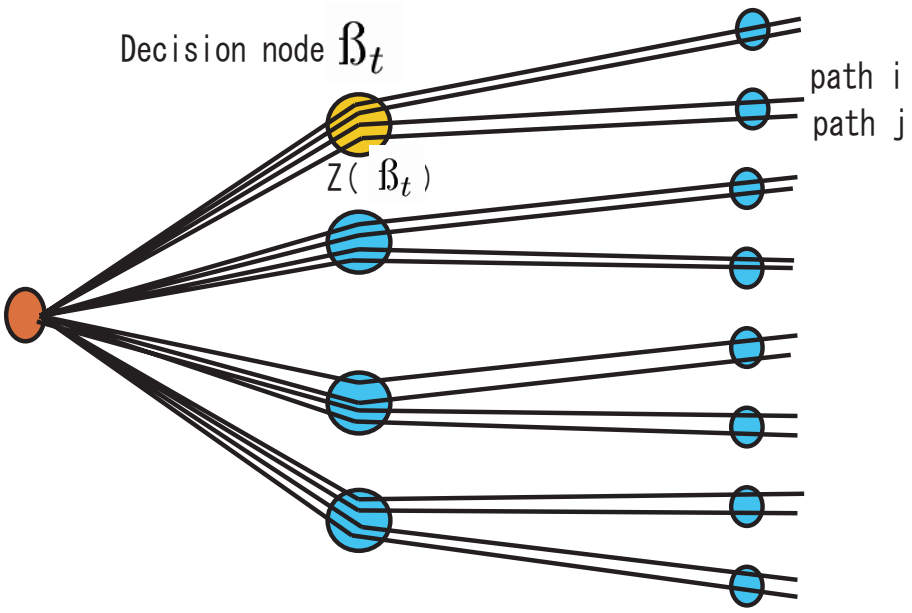

Figure 1: Simulation tree based on simulated sample paths

way to arrange sample paths so that the non-anticipativity condition holds. In our paper, taking into account a tree-like expanding decision space and the capability of making conditional decisions as in scenario tree models, we extend the approach by Hibiki [11, 12] to our portfolio optimization problem.

Moreover, we will show below that by treating risky and riskless assets separately, we can eliminate the decision variables of the riskless assets (*i.e.*, cash) from our compact reformulation model (see Appendix A for compact reformulation details). That indicates not only the simplicity but also the improvement of complexity of our model, which in turn reduces the computation time needed. Thus we will focus our attention mainly on risk assets in our paper from now on.

2.1 Construction of a Simulation Tree

We consider a T-period model where time ranges from $t = 0$ to $t = T$, and conditional decisions are taken at each time $t = 0, \ldots, T-1$. One riskless asset (cash) and J risky assets are available for portfolio selection. We let v_t and $P_t = (P_{t1}, \ldots, P_{tJ})', t = 0, \ldots, T$, denote the interest rate and risky asset prices vector, respectively, where prime means transposition. Clearly they are the only uncertainties in our model.

The possible future outcomes of interest rates and asset prices $(r_t, P_t')', t = 0, \ldots, T$, are revealed as I simulated sample paths

$$l^{(i)} = \left\{ \begin{pmatrix} r_0 \\ P_0 \end{pmatrix}, \begin{pmatrix} r_1 \\ P_1 \end{pmatrix}, \ldots, \begin{pmatrix} r_T^{(i)} \\ P_T^{(i)} \end{pmatrix} \right\}, \ i = 1, \ldots, I,$$

time. Figure (11) in Appendix B illustrates this setup

starting from the same initial state $(r_0\ P_0)'$ and traversing the entire planning horizon from $t = 0$ till $t = T$. At the beginning of the planning horizon, we assume that the initial state $(r_0\ P_0)'$ is known with certainty. At each time t along the planning horizon, all the sample paths bundled together at the previous period are further divided into some smaller bundles containing sample paths that exhibit similar characteristics, and different decisions are allowed to be made with respect to each bundle of sample paths. We repeat such process for $t = 1, \ldots, T-1$ and finally get a tree-like structure, which we call a simulation tree to distinguish it from a scenario tree.

The dissimilarity between two state points, corresponding to price vectors $P_t^{(i_1)}$ and $P_t^{(i_2)}$, is evaluated using Minkowski Distance Metric:

$$\text{dist } (P_t^{(i_1)}, P_t^{(i_2)}) = \{\textstyle\sum_{j=1}^{J} |P_{tj}^{(i_1)} - P_{tj}^{(i_2)}|^k\}^{1/k},$$

where k is a positive integer. In the case of $k = 2$, this metric gives the Euclidean distance. Clearly, the smaller this metric is, the more similar these two state points are to each other, indicating they could be bundled together and be applied with the same decision.

Then the simulation tree may be generated using a hierarchical clustering method. In fact, many different clustering methods have been proposed in the literature, such as the Nearest Neighbor method, Furthest Neighbor method, Group Average method, Ward method and so forth. Among these methods, we employ the Ward method, which is widely used and is believed to be superior to other methods for practice use; see [4, 19] for cluster analysis methods.

2.2 Identification of the Kernel of Each Bundle

Let B_t denote the set of all decision nodes at time t in a simulation tree, and Ω_t denote the whole possible state space at future time t. Let $ß_t \in B_t$ be one of decision nodes at time t and $V(ß_t)$ be the set of sample paths bundled together within the same decision node $ß_t$, *i.e.*, we use $i, j \in V(ß_t)$ to mean that both sample path i and sample path j, which are similar to one another, go through the same decision node $ß_t$.

Also, for each $ß_t$, we define $\Omega(ß_t) \subseteq \Omega_t$ as the sub-state space (or area) covering those scattered state points $P_t^{(i)}$ grouped in $ß_t$, where $i \in V(ß_t)$. Clearly, at each time t, we have

$$\bigcup_{ß_t \in B_t} V(ß_t) = \{1, \ldots, I\}, \tag{1}$$

$$\bigcup_{ß_t \in B_t} \Omega(ß_t) = \Omega_t. \tag{2}$$

Then the kernel $\Omega^K(ß_t)$ of a node $ß_t$ is defined as a subset of $\Omega(ß_t)$, *i.e.*, $\Omega^K(ß_t) \subseteq \Omega(ß_t) \subseteq \Omega_t$, which represents the area where most of the state points $\{P_t^{(i)}, i \in V(ß_t)\}$ are concentrated; for example, more than 95% of the state points in $V(ß_t)$. Denote

by $V^K(ß_t)$ the set of those most concentrated state points. Then the kernel may be determined as follows:

$$\Omega^K(ß_t) = \{\bar{P}^K(ß_t) + Hu| \parallel u \parallel \leq \gamma\} \subseteq \Omega(ß_t) \tag{3}$$

where $\bar{P}^K(ß_t) = \frac{1}{M}\sum_{i \in V^K(ß_t)} P_t^{(i)}$, M is the number of state points within the kernel and H is a symmetric positive definite matrix. The kernel $\Omega^K(ß_t)$ is an ellipse centered at $\bar{P}^K(ß_t)$, whose size is determined by γ.

Of course, one can also specify another percentile arbitrarily in his/her own definition of a kernel. To accomplish the definition of a kernel, we get two jobs to do: (i) First, identify the set of most concentrated state points according to the percentile specified; (ii) Then determine a proper matrix H so that the ellipse defined to be the kernel covers those most concentrated state points with as small volume (or area) as possible.

We use again the clustering method to choose the most concentrated state points within node $ß_t$. Then we determine the matrix H by $H = (\Sigma^K(ß_t))^{1/2}$, where $\Sigma^K(ß_t)$ is the covariance matrix of those state points selected. Finally, we adjust the value of γ so that the ellipse contains the specified percentile of state points in $ß_t$.

An alternative method to determine H is to solve the following convex optimization problem:

$$\begin{aligned}
\text{Min} \quad & -\log\det(X) && (4)\\
\text{s.t.} \quad & (P_t^{(i)} - \bar{P}^K(ß_t))^T X (P_t^{(i)} - \bar{P}^K(ß_t)) \leq 1,\ i \in V^K(ß_t) && (5)\\
& X \succeq \mathbf{0},\ X \in S^{J\times J}, && (6)
\end{aligned}$$

where $X \succeq \mathbf{0}$ means X is positive semidefinite and $S^{J\times J}$ denotes the set of $J \times J$ symmetric matrices. The objective function $-\log\det(X)$ represents the volume of the ellipse $\{P \in R^n | P^T X P \leq 1\}$, which is proved to be a convex function; see Fukushima [10]. Constraints (5) imply that all the state points in the kernel, $\{P_t^{(i)} | i \in V^K(ß_t)\}$, should be covered by this ellipse.

From our empirical experiments, we observe that for a randomly generated sample data set, the kernel defined by $\Sigma^K(ß_t)$,

$$\Omega^K(ß_t) = \{\bar{P}^K(ß_t) + (\Sigma^K(ß_t))^{\frac{1}{2}} u| \parallel u \parallel \leq \gamma\} \tag{7}$$

is similar to the kernel defined by the optimal solution X^* of (4)-(6),

$$\Omega^K(ß_t) = \{\bar{P}^K(ß_t) + (X^*)^{-\frac{1}{2}} u| \parallel u \parallel \leq 1\} \tag{8}$$

for some proper choice of γ.

In addition to a geometrical figure of the most concentrated state points, useful information about the covariance (or correlation) of those state points can be obtained as well. We give a visual illustration of determining a kernel using both an optimal solution X^* of (4)-(6) and covariance matrix $\Sigma^K(ß_t)$ with different value of γ in Figure 2. This figure shows 300 simulated future possible outcomes of

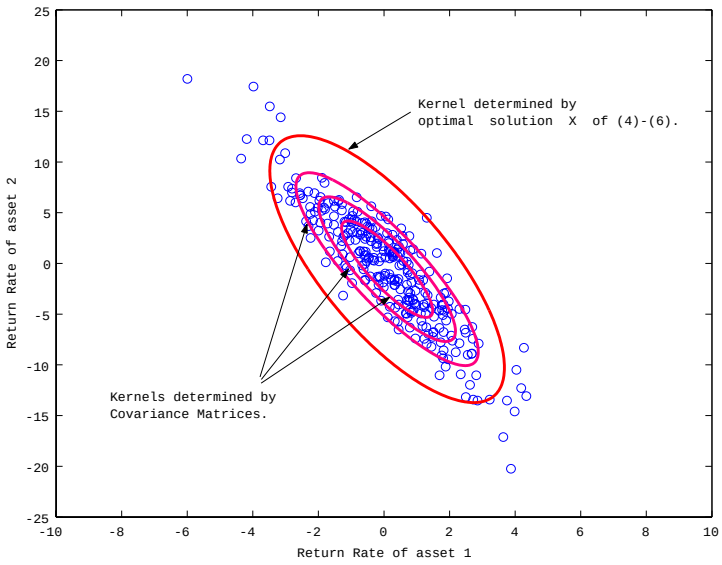

Figure 2: Two methods of construction a Kernel

return rate (r_1, r_2) and two different definitions of kernel, where r_i is the return rate of asset i with expected return $(\bar{r}_1, \bar{r}_2) = (-0.087\ 0.848)$, standard deviation $(\sigma_1, \sigma_2) = (0.78\ 5.571)$, and correlation $\rho_{12} = -0.101$. The inner ellipses correspond to the kernels defined by $\Sigma(K^{\beta_t})$ with $\gamma = 1, 1.5, 2$, see (8); While the outmost one corresponds to the kernel defined by X^* with $\gamma = 1$, see (9).

2.3 Modelling Procedure

In summary, the whole process of constructing a simulation tree and identifying a kernel of each node is stated as follows:

- Step 1: At time $t = 1$, a clustering method is implemented by using Minkowski Distance Metric with $k = 2$ as a dissimilarity measure between simulated sample paths and Ward Linkage[3] as a joining rule. We have some fixed-decision nodes whose number is predetermined. Different decisions are allowed to be applied to different decision nodes at time $t = 1$.

[3]Ward Linkage uses the incremental sum of squares; that is, the increase in the total within-group sum of squares as a result of joining groups r and s. It is given by

$$d(r,s) = \frac{n_r n_s}{n_r + n_s} d_{rs}^2,$$

where d_{rs}^2 is the distance between cluster r and cluster s, n_r and n_s are the numbers of objects in cluster r and s, respectively. The within-group sum of squares of a cluster is defined as the sum of the squares of the distance between all objects in the cluster and the centroid of the cluster.

- Step 2: Identify the kernel of each decision node generated at Step 1 using the method introduced in Section 2.2. Here we recommend Centroid Linkage[4] as a joining rule so that information about all of the sampled state points in each group could be used to define the joining.

- Step 3: Likewise, at time $t = 2, \ldots, T-1$, all sample paths that stem from the same decision node at time $t-1$ are further bundled together into some predetermined decision nodes using a clustering method. Also different decisions can be applied to different decision nodes at time t.

- Step 4: Repeat Step 2 and Step 3 until reaching time $T-1$. There is no need to cluster sampled paths at time T since no more decisions will be made at the end of the planning horizon.

3 Mathematical Formulations

Here we develop a T-period model, in which J risky assets and one riskless asset (such as cash) are available for the investment. Moreover, cash loan is not permitted in our model. At time t, decisions are made with respect to individual bundles of sample paths exhibiting similar characteristics.

3.1 Parameters

First, we define some parameters.

T: Number of periods;

J: Number of risky assets available for investment;

I: Number of simulation paths;

B_t: Set of bundles at time $t \in \{0, 1, \ldots, T\}$.

ß$_t$: Bundle in B_t, *i.e.*, ß$_t \in B_t$;

V(ß$_t$): Set of sample paths through bundle ß$_t$ at time t;

$P_0 = (P_{01}, \ldots, P_{0J})'$: Initial market price vector of risky assets, known with certainty;

$P_t^{(i)} = (P_{t1}^{(i)}, \ldots, P_{tJ}^{(i)})'$: Price vector of risky assets at time t along sample path i;

[4]Centroid Linkage uses the distance between the centroids of the two groups,

$$d(r, s) = d(\bar{x}_r, \bar{x}_s),$$

where $\bar{x}_r = \frac{1}{n_r} \sum_{i=1}^{n_r} x_{ri}$ and $\bar{x}_s$ is defined similarly; x_{ri} is the ith object in cluster r.

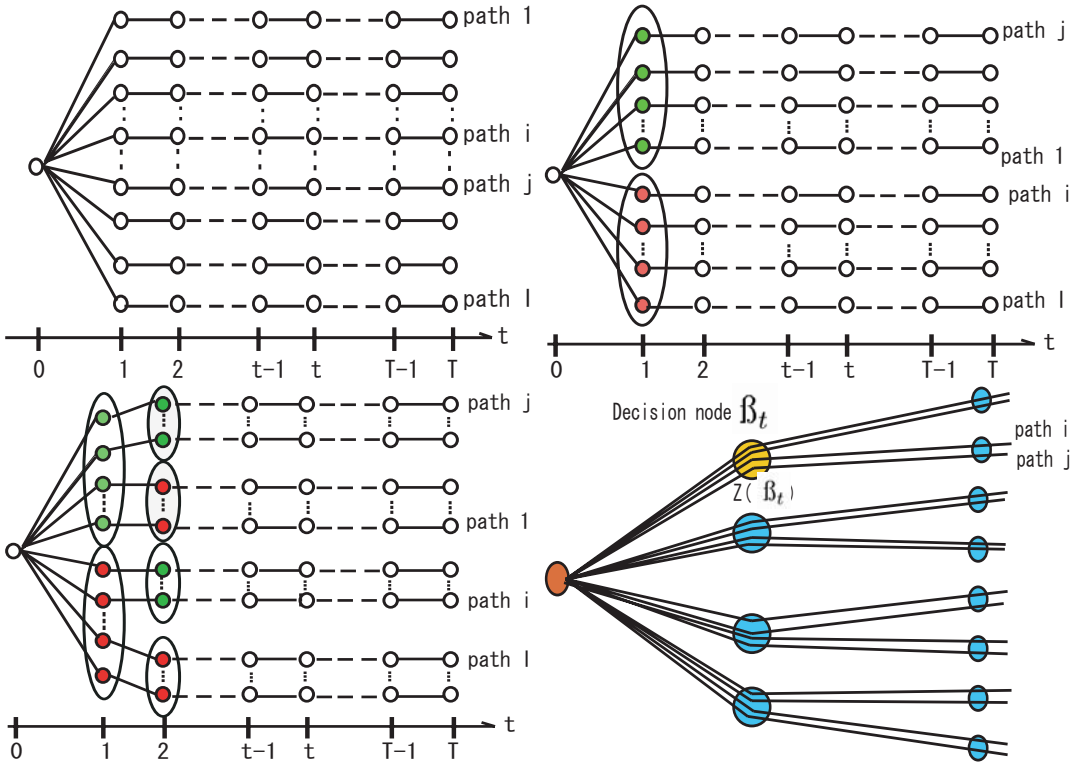

Figure 3: Example of constructing a 2-node simulation tree

r_0: Interest rate of cash (riskless asset) during the first period;

r_t: Interest rate of cash during period $[t, t+1], t = 1, \ldots, T-1$;

W_0: Total wealth at the beginning of the planning horizon;

W_G: Target wealth at the end of the planning horizon;

W_E: Required expected wealth at the end of the planning horizon.

3.2 Decision Variables

Next, we define decision variables for each time stage.

Variables at $t = 0$

At the beginning of the planning horizon, all information is assumed to be known with certainty.

$z(ß_0) = (z_1(ß_0), \ldots, z_J(ß_0))'$: Investment vector to risky assets for bundle $ß_0$;

v_0: Amount invested in the riskless asset.

Variables at $t = 1, \ldots, T-1$:

$z(ß_t) = (z_1(ß_t), \ldots, z_J(ß_t))'$:	Investment vector to risky assets for bundle $ß_t$;
$v_t^{(i)}$:	Cash investment along path i;
$q^{(i)}$:	Shortfall below the target wealth W_G at the end of the planning horizon along path i.

3.3 Objective Function and Model Formulation

The objective is to minimize the expectation shortfall, subject to budget constraints and safety constraints. In fact, because of the flexibility of the simulation model, many other regulation constraints (such as lower and/or upper bound constraints, full risk investment policy, transaction cost constraints, etc.) can also be incorporated in the formulation easily, even for each simulation path. Therefore the model can correspondingly be extended so as to be suitable for realistic problems. The problem is then formulated as a large-scale linear programming problem with a sparse coefficient matrix:

$$\text{Min} \quad \frac{1}{I}\sum_{i=1}^{I} q^{(i)} \tag{9}$$

s.t.

$$W_0 = (P_0)'z(ß_0) + v_0; \tag{10}$$

$$W_1^{(i)} = (P_1^{(i)})'z(ß_1) + v_1^{(i)} = (P_1^{(i)})'z(ß_0) + (1+r_0)v_0 \quad (i \in V(ß_1), ß_1 \in B_1); \tag{11}$$

$$W_t^{(i)} = (P_t^{(i)})'z(ß_t) + v_t^{(i)} = (P_t^{(i)})'z(ß_{t-1}) + (1+r_{t-1}^{(i)})v_{t-1}^{(i)} \quad (t = 2, \ldots, T-1; i \in V(ß_k), ß_k \in B_k, k = t-1, t); \tag{12}$$

$$W_T^{(i)} = (P_T^{(i)})'z(ß_T) + (1+r_{T-1}^{(i)})v_{T-1}^{(i)} \quad (i \in V(ß_{T-1}), ß_{T-1} \in B_{T-1};); \tag{13}$$

$$W_T^{(i)} + q^{(i)} \geq W_G \ (i = 1, \ldots, I); \tag{14}$$

$$\frac{1}{I}\sum_{i=1}^{I} W_T^{(i)} \geq W_E; \tag{15}$$

$$z(ß_0) \geq 0; \tag{16}$$

$$z(ß_t) \geq 0 \ (ß_t \in B_t; t = 1, \ldots, T-1); \tag{17}$$

$$v_0 \geq 0; \tag{18}$$

$$v_t^{(i)} \geq 0 \ (t = 1, \ldots, T-1; i = 1, \ldots, I); \tag{19}$$

$$q^{(i)} \geq 0 \ (i = 1, \ldots, I). \tag{20}$$

Although this formulation is easy to understand, a compromise still has to be made between describing the precision of the underlying uncertainty and making conditional decisions with relatively high flexibility. Taking a closer look at the

constraints of the formulation, we find that the number of decision variables $z(ß_t)$ that represent the investment vector depends only on the number of bundles in the decision tree, while the number of decision variables $v_t^{(i)}$ that represent the investment quantities in riskless assets depends on both the number of simulated sample paths and the number of periods. Therefore, by treating risk and riskless assets separately, the number of decision variables can be decreased substantially, thereby the problem size and the computational time can also be reduced. Another reason for treating the risky assets and riskless assets separately is that their totally different instinct properties. That is, the return rate for a riskless asset is already determined at the beginning of each period; while the return rate for a risky asset (such as stock, bond, etc.) remains unknown until the end of that period.

The equality constraints (10)-(13) in the above formulation may be rewritten as inequality constraints by eliminating the cash variables. However, this does not mean that the cash variables are excluded from the asset allocation decision. Instead, cash variables emerge implicitly in the resulting compact formulation and can be calculated using risk asset variables. After elimination of the cash variables, the original formulation can be rewritten as follows:

$$\text{Min} \quad \frac{1}{I}\sum_{i=1}^{I} q^{(i)} \tag{21}$$

$$\text{s.t.} \quad (P_0)'z(ß_0) \le W_0; \tag{22}$$

$$(P_1^{(i)})'z(ß_1) - (\eta_1^{(i)} * P_0)'z(ß_0) \le (1+r_0)W_0 \quad (i \in V(ß_1), ß_1 \in B_1); \tag{23}$$

$$(P_t^{(i)})'z(ß_t) - (\eta_t^{(i)} * P_{t-1}^{(i)})'z(ß_{t-1}) - (1+r_{t-1}^{(i)})(\eta_{t-1}^{(i)} * P_{t-2}^{(i)})'z(ß_{t-2}) - \ldots - \prod_{k=1}^{t-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) \le \prod_{k=0}^{t-1}(1+r_k^{(i)})W_0 \quad (i \in V(ß_1)\bigcap\ldots\bigcap V(ß_k), ß_k \in B_k, k=0,\ldots,t; t=2,\ldots,T-1); \tag{24}$$

$$-q^{(i)} - (\eta_T^{(i)} * P_{T-1}^{(i)})'z(ß_{T-1}) - (1+r_{T-1}^{(i)})(\eta_{T-1}^{(i)} * P_{T-2}^{(i)})'z(ß_{T-2}) - \ldots - \prod_{k=1}^{T-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) \le \prod_{k=0}^{T-1}(1+r_k^{(i)})W_0 - W_G \quad (i \in V(ß_k), ß_k \in B_k, k=0,\ldots,T-1); \tag{25}$$

$$\frac{1}{I}\sum_{i=1}^{I} W_T^{(i)} \ge W_E; \tag{26}$$

$$z(ß_t) \ge 0 \ (ß_t \in S_t; t=0,\ldots,T-1); \tag{27}$$

$$q^{(i)} \ge 0 \ (i=1,\ldots,I), \tag{28}$$

where $*$ denotes the array multiplication operator, or Hadamard product[5] of two

[5]For $x=(x_1,\ldots,x_n)'$ and $y=(y_1,\ldots,y_n)'$, the Hadamard product of x and y is defined as $x*y=(x_1y_1,\ldots,x_ny_n)'$.

vectors. Assuming $\mu_{tj}^{(i)}$ is the return rate of asset j during period $[t-1,t]$ along sample path i, we interpret $\eta_t^{(i)} = (\eta_{t1}^{(i)}, \ldots, \eta_{tJ}^{(i)})'$ as the net profit rates given by

$$\begin{aligned} \eta_{tj}^{(i)} &= p_{tj}^{(i)} - (1 + r_{t-1}^{(i)}) p_{t-1j}^{(i)} \\ &= (1 + \mu_{tj}^{(i)}) p_{t-1j}^{(i)} - (1 + r_{t-1}^{(i)}) p_{t-1j}^{(i)} \\ &= (\mu_{tj}^{(i)} - r_{t-1}^{(i)}) p_{t-1j}^{(i)} \ (j = 1, \ldots, J). \end{aligned} \tag{29}$$

Although the problem size will grow quickly if we increase the number of simulation paths in order to obtain a detailed description of the underlying uncertainty, the coefficient matrix of the problem remains sparse (see Figure 4). Many advanced mathematical programming software packages such as SeDuMi [25] can handle large-scale optimization problems efficiently by exploiting sparsity of the coefficient matrix. Moreover, from the experimental results as illustrated in Figure 5, the original formulation tends to require much more computational time than the compact formulation as the number of sample paths increases.

4 Risk Chance Constraints

Taking into account the stability of the optimal solutions as well as the shortfall risk at each period, two types of risk chance constraints will be discussed in this section.

The common belief that high risk leads to high return is the very motivation driving people to invest in risky assets instead of saving their fortune in a bank. Thus we will reflect this widely held option in our model as well.

At the beginning of period $[t-1,t]$, the current and the previous market information and investment history are supposed to be known with certainty. The only uncertainty is the market price at time t, which determines the return corresponding to the decisions made at time $t-1$. We require that at the end time t of this period, for most cases the investor who takes the risk and aggressively invests in risky assets will get much more profit than a conservative investor who just saves his money in a bank to hedge risk at the beginning of the period; for instance, during the period $[t-1,t]$, we require that

$$\vartheta = W_t - (1 + r_{t-1}) W_{t-1} \geq a \text{ with high probability}, \tag{30}$$

for some constant $a > 0$, where W_t is the total wealth at time t and r_{t-1} is the interest rate during period $[t-1,t]$. Note that $(1 + r_{t-1})W_{t-1}$ represents the total wealth obtained by a conservative investor. We call constraint (30) a Risk Chance Constraint. The choice of a in our model reflects the investor's attitude toward the compromise between risk and profit. Since

$$W_t = P_t' z_t + v_t = P_t' z_{t-1} + (1 + r_{t-1}) v_{t-1}, \tag{31}$$

ϑ in (30) can be simplified as follows:

$$\vartheta = W_t - (1 + r_{t-1}) W_{t-1}$$

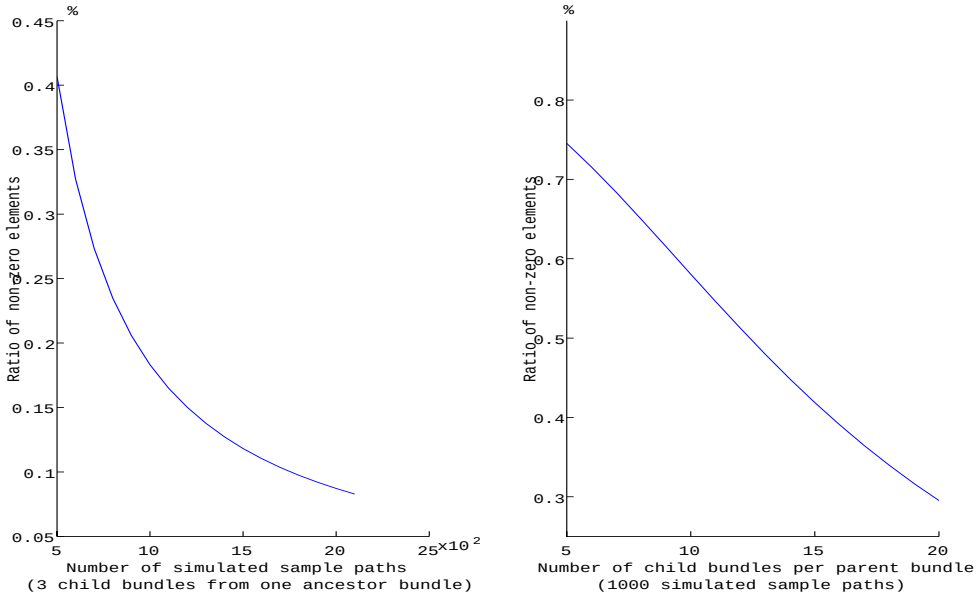

Figure 4: Sparsity of the coefficient matrix

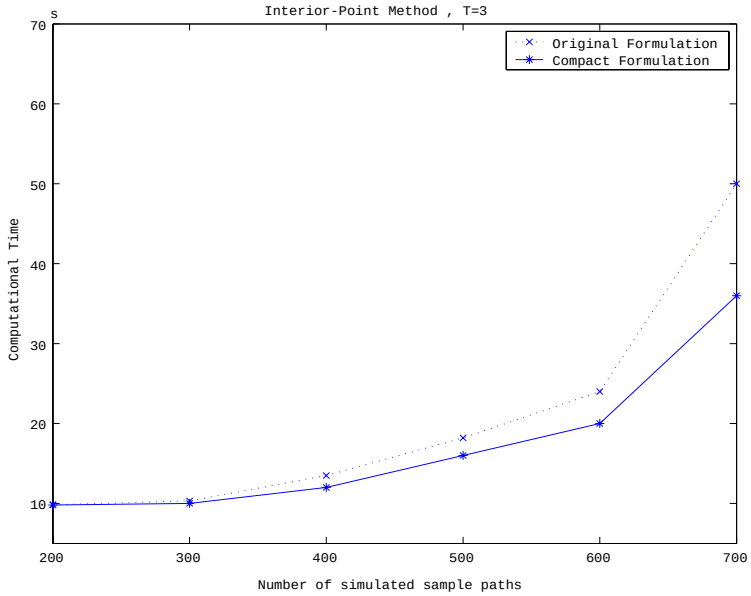

Figure 5: Comparison of the computational time for original formulation and compact formulation. Implemented by using *fmincon* solver of Matlab R13.

$$
\begin{aligned}
&= P_t' z_{t-1} + (1 + r_{t-1}) v_{t-1} - (1 + r_{t-1})(P_{t-1}' z_{t-1} + v_{t-1}) \\
&= (P_t - (1 + r_{t-1}) P_{t-1})' z_{t-1}. \qquad (32)
\end{aligned}
$$

Two different discrete versions of Risk Chance Constraints are proposed in our paper. The notable distinction between them is that the target objects to which Risk Chance Constraints will be applied are different. In the first case, Risk Chance Constraint is imposed to the kernel $\Omega^K(ß_t)$ of node $ß_t$, while in second case, Risk Chance Constraint is imposed directly to the set $V(ß_t)$ of all sample paths going through node $ß_t$.

4.1 Risk Chance Constraint (I)

Let $\omega(ß_t)$ denote the parent node of $ß_t$. Then Risk Chance Constraint (30) can be formulated as follows: For each decision node $ß_t \in B_t, t = 1, \ldots, T$, and constant $a > 0$, we have

$$
\vartheta = (P(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) \geq a, \qquad (33)
$$

where $\overline{r}(\omega(ß_t)) = \frac{1}{M} \sum_{i \in V(\omega(ß_t))} r_{t-1}^{(i)}, \overline{P}(\omega(ß_t)) = \frac{1}{M} \sum_{i \in V(\omega(ß_t))} P_{t-1}^{(i)}$, and M is the number of sample paths passing through $\omega(ß_t)$.

For any market risky asset price vector $P(ß_t)$ within the kernel of decision node $ß_t$, *i.e.*,

$$
P(ß_t) \in \Omega^K(ß_t) = \{\bar{P}^K(ß_t) + Hu | \parallel u \parallel \leq \gamma\}, \qquad (34)
$$

we require the value of ϑ be greater than a, which reads as follows:

$$
(P(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) \geq a \text{ for all } P(ß_t) \in \Omega^K(ß_t). \qquad (35)
$$

We can show that such a constraint can be cast as an SOC constraint. In fact, since

$$
\begin{aligned}
& \min \{(P(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) | P(ß_t) \in \Omega^K(ß_t)\} \qquad (36) \\
= \; & \min \{(\bar{P}^K(ß_t) + Hu - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) | \parallel u \parallel \leq \gamma\} \qquad (37) \\
= \; & (\bar{P}^K(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) + \min_{\|u\| \leq \gamma} (Hu)' z(\omega(ß_t)) \qquad (38) \\
= \; & (\bar{P}^K(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) - \gamma \parallel Hz(\omega(ß_t)) \parallel, \qquad (39)
\end{aligned}
$$

(35) is equivalent to the inequality

$$
\gamma \parallel Hz(\omega(ß_t)) \parallel \leq (\bar{P}^K(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) - a. \qquad (40)
$$

This is an SOC constraint.

4.2 Risk Chance Constraint (II)

Assuming that the asset prices or the return rates have a jointly Gaussian distribution, we may consider an alternative risk chance constraint in our model.

At decision making time point $t-1$, except the price vector P_t, the price vectors $P_{[t]} = \{P_0, \ldots, P_{t-1}\}$ and decision variable vectors $z_{[t]} = \{z_0, \ldots, z_{t-1}\}$ at the previous periods are supposed to be known with certainty. For period $[t-1, t]$, the risk chance constraint can also be expressed by

$$\Pr\{\vartheta \geq a | P_{[t]}, z_{[t]}\} = \Pr\{(P(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) \geq a | P_{[t]}, z_{[t]}\} \geq \beta, \quad (41)$$

or equivalently,

$$\Pr\{(P(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) \leq a | P_{[t]}, z_{[t]}\} \leq 1 - \beta. \quad (42)$$

We will show that such a probability constraint can also be cast as an SOC constraint.

Letting $X(ß_t) = (P(ß_t) - (1+\overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t))$, the mean and variance of $X(ß_t)$ can be obtained as follow:

$$\begin{aligned} \overline{X}(ß_t) &= E(X(ß_t)) = (\overline{P}(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)), & (43) \\ \sigma^2(ß_t) &= V(X(ß_t)) = V\{(P(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t))\} \\ &= V\{(P(ß_t))' z(\omega(ß_t))\} = (z(\omega(ß_t))^T \Sigma(ß_t) z(\omega(ß_t)). & (44) \end{aligned}$$

Here, $\overline{P}(ß_t)$ and $\Sigma(ß_t)$ are the mean price vector and the covariance matrix of all state points belonging to the decision node $ß_t$.

Therefore, the previous constraint can be converted as follows:

$$\begin{aligned} & \Pr(X(ß_t) \leq a) \leq 1 - \beta & (45) \\ \Leftrightarrow \quad & \Pr\left(\frac{X(ß_t) - \overline{X}(ß_t)}{\sigma(ß_t)} \leq \frac{a - \overline{X}(ß_t)}{\sigma(ß_t)}\right) \leq 1 - \beta \\ \Leftrightarrow \quad & \Phi\left(\frac{a - \overline{X}(ß_t)}{\sigma(ß_t)}\right) \leq 1 - \beta \\ \Leftrightarrow \quad & a - \overline{X}(ß_t) \leq \Phi^{-1}(1 - \beta)\sigma(ß_t) \\ \Leftrightarrow \quad & a \leq (\overline{P}(ß_t) - (1 + \overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) \\ & + \Phi^{-1}(1 - \beta) \parallel (\Sigma(ß_t))^{1/2} z(\omega(ß_t)) \parallel, & (46) \end{aligned}$$

where

$$\Phi(b) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{b} \exp^{-\frac{t^2}{2}} dt$$

is the CDF of a zero mean unit variance Gaussian random variable. Now, provided that $\beta \geq 1/2$, *i.e.*, $\Phi^{-1}(1 - \beta) \leq 0$, constraint (46) is a SOC constraint.

However, as pointed out by Boyd et al. [16], in practice the observed returns are seldom Gaussian. They often are skewed, or have fat tails, *i.e.*, resemble a Gaussian distribution in the central area but have higher probability mass for high deviations. Nevertheless, a shortfall probability approach can be effective if used in an informed manner. Alternatively, one can use the Chebyshev bound to limit the shortfall probability, as suggested by Roy [22]. In this case, the factor $\Phi^{-1}(1 - \beta)$ in the formulas above is replaced by $(\beta)^{-1/2}$.

5 An Alternative Risk Measurement: CVaR

So far, we have been using the first-order lower partial moment (LPM_1)[6], a downside risk measure, of the terminal wealth as the risk measure. It is of great importance for those people who intend to work in the finance field to know such downside risk measures by the following reasons: (i) Since individuals typically are loss averse and not gain averse, an asymmetric risk measure is always preferred; and (ii) Regulation authorities recommend the use of a downside risk measure Value-at-Risk (VaR). However, a key missing property of VaR is convexity, which means many local optimum may occur and a global optimal solution is hardly computable. Hence, such a risk measure is not appropriate for practical portfolio optimization problems.

Here we will extend our model and combine it with a new risk management technique using Conditional Value-at-Risk (CVaR), see Rockafellar and Uryasev [21]. The CVaR is one of downside risk measures as well and it turns out to be very powerful and economically sound for portfolio optimization problems because of its convexity that the crucial property VaR lacks, even though it was first defined as the weighted average of VaR and the loss exceeding VaR.

Other reasons for our choice of CVaR are listed as follows:

- CVaR is a subadditive measure of risk (see Pflug [20] and Rockafellar and Uryasev [21]), implying diversification of a portfolio reduces CVaR.
- CVaR is a coherent measure of risk in the sense of Artzner et al. [2], and Acerbi and Tasche [1].
- CVaR does not destroy the convexity structure of our model and can be easily dealt with numerically by means of the Monte Carlo simulation method.
- CVaR can be obtained easily without involving expensive calculation of VaR (see [27] for details).

We say we have a loss along a sample path i at the end of the planning horizon if $W_T^{(i)}$ is less than W_G. To prevent terminal wealth shortfall risk and to improve the performance, we introduce the concept of CVaR into our model and replace constraints (14) and (20) with the following one:

$$W_T(z_{[T]}) \geq W_G \text{ with high probability (confidence level) } \alpha, \tag{47}$$

which indicates that terminal wealth of $100\alpha\%$ of the sample path set should be equal or greater than W_G. It sounds much more reasonable for a portfolio manager or an investor.

As a measure of loss, we use the difference between the terminal wealth $W_T(z_{[T]})$ and the postulated target wealth W_G:

$$h(z_{[T]}) = W_G - W_T(z_{[T]}).$$

[6] $LPM_k = \frac{1}{I}\sum_{i=1}^{I} |W_T^{(i)} - W_G|_-^k$, where k is a positive integer, $W_T^{(i)}$ is the terminal wealth of each path i, W_G is the target wealth predetermined at the beginning of the planning horizon, and $|\cdot|_-$ denotes $\min\{\cdot, 0\}$.

Thus constraint (47) can be restated as

$$h(z_{[T]}) \leq 0 \text{ with high probability (confidence level) } \alpha. \tag{48}$$

Let Q be the joint probability measure of $P_{[T]} = \{P_0, \ldots, P_{T-1}\}$, and denote by $\Phi(\xi)$ the cumulative probability distribution of the loss $h(z_{[T]})$:

$$\Phi(\xi) = Pr(h(z_{[T]}) \leq \xi) = \int_{h(z_{[T]}) \leq \xi} Q(dP_{[T]}), \tag{49}$$

which by definition is the probability that the loss $h(z_{[T]})$ does not exceed a threshold value ξ. Then constraint (48) can further be restated as follows::

$$\xi_\alpha \leq 0, \text{ where } \xi_\alpha = \min\{\xi \in R | \Phi(\xi) \geq \alpha\}. \tag{50}$$

The value of ξ_α is called α-Value at Risk (α-VaR) and (50) means that the loss in at least $100\alpha\%$ of outcomes must be below or equal to 0. (Note that, in general, this threshold level may be chosen to be different from 0.) Because of several notable drawbacks of VaR, we proceed to discuss Conditional Value-at-Risk (CVaR), which has much better theoretical properties as mentioned earlier. See Figure 12 in the Appendix C for illustration.

There are several definitions of CVaR in the literature. For a better understanding of this important risk measure as well as its applications in our model, we will just briefly introduce two general definitions below; for other definitions and applications in finance field, see Vanini and Vignola [27], and Uryasev [26].

- The first definition:

  $$\text{CVaR}_\alpha = E[h(z_{[T]}) | h(z_{[T]}) \geq \xi_\alpha].$$

  This means that, given the value of ξ_α determined by (49) and (50), CVaR_α is the conditional expectation of loss exceeding ξ_α.

- The second definition (a discrete version of CVaR_α):

  $$\text{CVaR}_\alpha = \xi_\alpha + \frac{1}{(1-\alpha)I} \sum_{i=1}^{I} [h(z_{[T]}^{(i)}) - \xi_\alpha]^+$$

  where, $h(z_{[T]}^{(i)})$ is the loss along path i,

  $$h(z_{[T]}^{(i)}) = W_G - W_T(z_{[T]}^{(i)})$$

  and $[\cdot]^+ = \max\{\cdot, 0\}$.

It is clear that CVaR always exceeds or equals VaR, *i.e.*, $\text{VaR}_\alpha \leq \text{CVaR}_\alpha$, see [26]. In our model, we replace (48) by the CVaR constraint

$$\phi_\alpha = \text{CVaR}_\alpha \leq \tau, \tag{51}$$

where τ can be chosen appropriately. When $\tau = 0$, we have a risk constraint that dominates, *i.e.*, is stronger than, the α-VaR constraint (48). Using a negative τ would tighten the constraint further, while a positive one would relax it.

For the simulation-based method, as is shown by Rockafellar and Uryasev [21], constraint (51) can be replaced by linear constraints:

$$\xi_\alpha + \frac{1}{(1-\alpha)I}\sum_{i=1}^{I} q^{(i)} \leq \tau \tag{52}$$

$$W_G - W_T^{(i)} - \xi_\alpha \leq q^{(i)} \tag{53}$$

$$q^{(i)} \geq 0,\ i = 1, \ldots, I,$$

where $q^{(i)}, i = 1, \ldots, I$ are dummy variables. If constraint (52) is active at an optimal solution, the corresponding optimal value of ξ, if it is unique, will be equal to VaR. If there are many optimal values of ξ, then VaR is the left endpoint of the optimal interval. The left-hand side of (52) will be equal to CVaR.

The optimization problem incorporated with CVaR is presented as

$$\text{Max} \quad \frac{1}{I}\sum_{i=1}^{I} W_T^{(i)} \tag{54}$$

$$\text{s.t.} \quad \xi_\alpha + \frac{1}{(1-\alpha)I}\sum_{i=1}^{I} q^{(i)} \leq \tau; \tag{55}$$

$$W_G - W_T^{(i)} - \xi_\alpha \leq q^{(i)}; \tag{56}$$

$$q^{(i)} \geq 0, i = 1, \ldots, I;$$

$$\text{and (22),(23),(24),(27),} \tag{57}$$

or

$$\text{Min} \quad \xi_\alpha + \frac{1}{(1-\alpha)I}\sum_{i=1}^{I} q^{(i)} \tag{58}$$

$$\text{s.t.} \quad \frac{1}{I}\sum_{i=1}^{I} W_T^{(i)} \geq W_E; \tag{59}$$

$$W_G - W_T^{(i)} - \xi_\alpha \leq q^{(i)}; \tag{60}$$

$$q^{(i)} \geq 0, i = 1, \ldots, I;$$

$$\text{and (22),(23),(24),(27),} \tag{61}$$

in which the number of decision variables is increased by 1.

Since the additional CVaR constraints do not destroy the linear structure of the problem, the solution set is still a convex polyhedron, which is amenable to the computation of a global optimal solution. More specifically, the final problem is a large-scale LP problem and can be solved using any efficient LP algorithm.

In summary, we can integrate the original proposed hybrid simulation method with risk chance constraints or CVaR to prevent the shortfall risk as much as possible while keeping the tractability of the problem at the same time.

Furthermore we can apply both of these two ideas, risk chance constraints and CVaR, to obtain the following integrated formulation:

$$\text{Min} \quad \xi_\alpha + \frac{1}{(1-\alpha)I}\sum_{i=1}^{I} q^{(i)} \tag{62}$$

$$\text{s.t.} \quad P_0' z(ß_0) \le W_0; \tag{63}$$

$$(P_1^{(i)})' z(ß_1) - (\beta_1^{(i)} * P_0)' z(ß_0) \le (1+r_0)W_0 \tag{64}$$
$$(i \in V(ß_1), ß_1 \in B_1);$$

$$(P_t^{(i)})' z(ß_t) - (\beta_t^{(i)} * P_{t-1}^{(i)})' z(ß_{t-1}) - (1+r_{t-1}^{(i)})(\beta_{t-1}^{(i)} * P_{t-2}^{(i)})' z(ß_{t-2}) - \ldots$$
$$- \prod_{k=1}^{t-1}(1+r_k^{(i)})(\beta_1^{(i)} * P_0)' z(ß_0) \le \prod_{k=0}^{t-1}(1+r_k^{(i)})W_0 \tag{65}$$
$$(t = 2, \ldots, T-1; i \in V(ß_k), ß_k \in B_k, k = 0, \ldots, t);$$

$$\frac{1}{I}\sum_{i=1}^{I} W_T^{(i)} \ge W_E; \tag{66}$$

$$W_G - W_T^{(i)} - \xi_\alpha \le q^{(i)} \ (i = 1, \ldots, I); \tag{67}$$

$$\gamma \parallel Hz(\omega(ß_t)) \parallel \le (\bar{P}^K(ß_t) - (1+\overline{r}(\omega(ß_t)))\overline{P}(\omega(ß_t)))' z(\omega(ß_t)) - a \tag{68}$$
$$(t = 0, \ldots, T-1; ß_t \in B_t);$$

$$q^{(i)} \ge 0 \ (i = 1, \ldots, I); \tag{69}$$

$$z(ß_t) \ge 0 \ (ß_t \in B_t; t = 0, \ldots, T-1). \tag{70}$$

6 Numerical Experiments

We test our proposed model using a 3-period portfolio selection problem. It consists of 1000 sample paths. Even though theoretically the model can handle quite a large number of financial asset categories, here to simplify interpretation of the results, we have mostly limited ourselves to only four asset categories: cash, stock, bond and CB. Furthermore, without loss of generality, initial prices of stock, bond and CB are assumed to be 1.

To focus mainly on evaluating the performance of CVaR and Risk Chance Constraints, we only incorporated some basic budget constraints, CVaR and Risk Chance Constraints (I) in our numerical experiments. For realistic application, other kinds of constraints, *i.e.*, full investment constraints, transaction cost constraints, special regulation constraints can also be added in our model easily.

6.1 Data and Computational Resource

The data set of sample paths was generated using statistic data provided by Nikko Inc., see Table 1.

Table 1: Experimental Data[7]

Correlation Matrix		Cash			Stock			Bond			CB		
		1	2	3	1	2	3	1	2	3	1	2	3
	1	1	-.091	.073	-.101	0	-.030	-.238	.008	.09	-.146	-.044	-.052
Cash	2	-.091	1	-.092	.045	-.094	-.007	-.183	-.237	.011	-.012	-.144	-.047
	3	.073	-.092	1	.016	.042	-.091	-.166	-.188	-.221	-.062	-.017	-.138
	1	-.101	.045	.016	1	.022	-.031	.145	-.173	-.096	.761	.042	-.045
Stock	2	0	-.094	.042	.022	1	.018	.085	.144	-.17	.019	.76	.041
	3	-.032	-.007	-.091	-.031	.018	1	.077	.085	.141	.011	.019	.76
	1	-.238	-.183	-.166	.145	.085	.077	1	.13	-.108	.327	.202	.065
Bond	2	.008	-.237	-.188	-.173	.144	.085	.13	1	.137	-.114	.327	.204
	3	.090	.011	-.221	-.096	-.17	.141	-.108	.137	1	-.18	-.109	.321
	1	-.146	-.012	-.062	.761	.019	.011	.327	-.114	-.18	1	.092	-.068
CB	2	-.044	-.144	-.017	.042	.76	.019	.202	.327	-.109	.092	1	.093
	3	-.052	-.047	-.138	-.045	.041	.76	.065	.204	.321	-.068	.093	1
Expected Return		-.087	-.081	-.089	.848	.867	.843	.625	.623	.645	.786	.78	.786
Standard Deviation		.78	.784	.778	5.571	5.582	5.595	1.372	1.372	1.353	3.543	3.541	3.538

The return rate $\mu_{jt}^{(i)}$ is calculated as follows:

$$\mu_{jt}^{(i)} = \bar{\mu}_{jt} + \sigma_{jt}\varepsilon_{jt}^{(i)},\ j = 0, \ldots, J;\ t = 1, \ldots, T, \tag{71}$$

where $\bar{\mu}_{jt}$ and σ_{jt} are expected return and standard deviation, respectively, and $\varepsilon_{jt}^{(i)}(j = 0, \ldots, J; t = 1, \ldots, T)$ are random numbers of multi-variate standardized normal distribution such that

$$\varepsilon_{jt}^{(i)} \sim N(\mathbf{0}, \Sigma), \tag{72}$$

where Σ is the $(J+1)T \times (J+1)T$ correlation matrix given by Table 1.

Clearly, from the formulation of our model, we can see that the problem size will grow linearly as the number of sample paths increases. On the other hand, in order

[7] These are the Nikko Stock performance index (TSE1), Nikko bond performance index, Nikko CB performance index and call rate, provided by Nikko Cordial Securities Inc. Also they have been used in Hibiki [12].

to keep a sufficiently high approximation accuracy, the number of sample paths required is determined mainly by the quality of random numbers $\varepsilon_{jt}^{(i)}$. In other words, using high quality random numbers with low-discrepancy, such as quasi-random numbers, will help us control the problem size. Thus, in our numerical experiments, we use random numbers generated by NtRand [8] to simulate the set of sample paths. To achieve convergence, various moment matching methods have been adopted in NtRand. It is said that NtRand is a very efficient and speedy tool for generating random numbers, especially multi-variate random numbers with complex correlations. Therefore, it has been widely used since late 1990s in both academic research fields and real business practices. It only took NtRand about 1 second to generate 12,000 random numbers for our experiments.

Interest rate $r_t^{(i)}$ of each period can be obtained by

$$r_1^{(i)} = r_0 \times (1 + \mu_{01}^{(i)}), \tag{73}$$

$$r_t^{(i)} = r_{t-1}^{(i)} \times (1 + \mu_{0t}^{(i)}), \tag{74}$$

where $r_0 = 0.404\%$ is the initial interest rate.

A full list of default parameters is as follows:

Investment period: $T = 3$;

Number of asset categories: $J = 4$;

Number of sample paths: $I = 1000$;

Initial wealth: $W_0 = 10000$;

Initial interest rate, $r_0 = 0.404\%$;

Terminal target wealth for each sample path: $W_G = W_0 = 10000$;

Percentage used to defined kernel: 85%;

Constant a, used to define Risk Chance Constraints: $a = 0$;

Confidence level α, used to define CVaR: $\alpha = 0.9$.

We implement our model (63)-(72) using Matlab R13 on the platform of Windows XP with 512MB of RAM. The computation time needed for problems with 1000 sample paths, 4-node simulation tree is about 260 seconds.

6.2 Computational Results

The efficient frontier in Figure 6 shows that an investor will have to take much more risk if he/she prefers a higher expectation of terminal wealth return W_E. This numerical result is consistent with the widely held option among the real practice field: high risk leads to high return.

[8] NtRand (Numerical Technologies Random Generator for Excel) is a free add-in EXCEL program developed by Numerical Technologies, Inc. NtRand and its manual can be downloaded from http://www.numtech.co.jp/NtRand/.

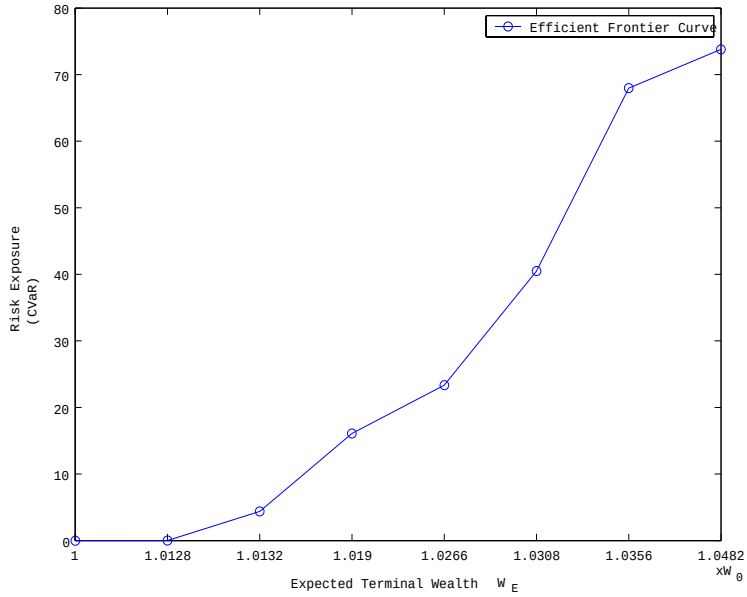

Figure 6: Efficient frontier of Expected Terminal Wealth Return W_E vs. CVaR

For an investor trying to find a sequence of optimal investment solutions by solving a multi-period optimal problem like ours, what he/she cares most is the question: What should I do now? Since the real world is often different from what we can only roughly predict in advance, there is no need at all for an investor to do exactly according to the optimal solution obtained by obsolete information. He/She can solve the problem again with new available information so as to achieve a much more suitable new sequence of optimal investment solutions. Thus, the initial investment portfolio is of the greatest importance to our model users.

Figure 7 illustrates initial investment compositions for different expected terminal wealth return W_E. When an investor has a relatively moderate expectation for terminal wealth return, such as $W_E \leq 1.0128 \times W_0$, more than 87% of initial wealth is invested in cash, the safest asset among all available assets, and no investment is made to high risk assets (stock, CB) at all. As the expectation of terminal wealth return continues to grow up, he/she has to decrease his/her investment in safe asset (cash) and invest more money to relatively high risk assets to accomplish his/her ambitious desire. Finally, as the expectation of terminal wealth return reaches $W_E = 1.0482 \times W_0$, he/she has to invest all his/her wealth in risk assets completely from the very beginning of the planning horizon.

We also examined our model performance when the number of child nodes stemming from the same parent node changes. Consistent with the case of scenario tree model, in Figure 8, our results show that for a specific terminal wealth expectation, the more the number of child nodes is, the less the risk will be. The reason for this phenomenon is that an investor can have more flexibility of conditional decision making and therefore can choose whether to invest more money into high risk assets

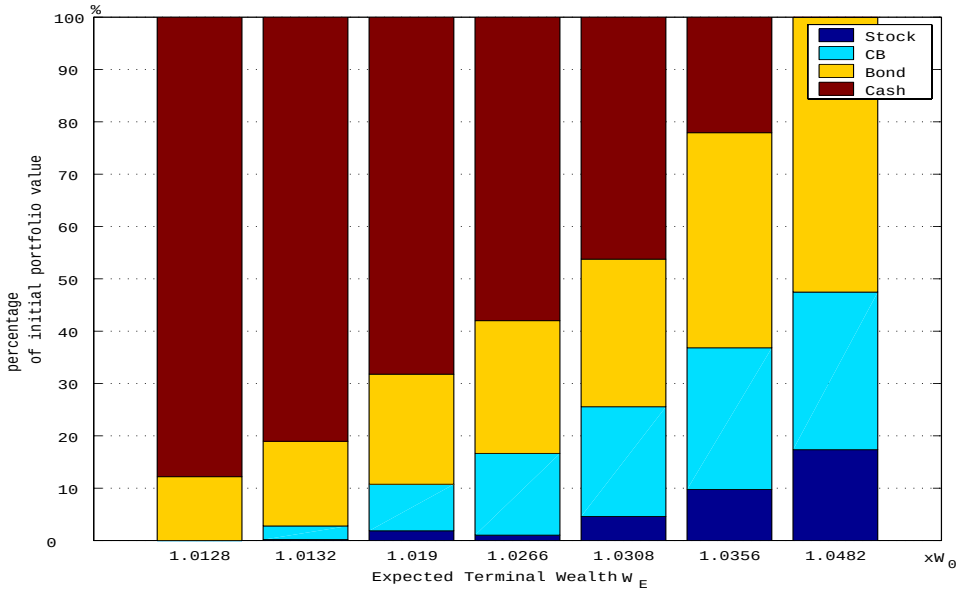

Figure 7: Initial portfolio compositions for different cases.

or not, according to the changes of asset prices.

As illustrated in Figure 9, optimal values obtained using either Ward clustering method or Average clustering method, two often used clustering methods, show a convergence behavior when the number of simulated sample paths increases. Even though they did not converge upon the same risk value due to different simulation tree structures they lead to, our model users can still take advantage of such a convergence property so as to limit the problem size and obtain a sequence of optimal solutions quickly.

Beyond that, we observe that different percentages used to define the kernel of a bundle cause different convergence speed when the number of simulated sample paths increases, see Figure 10.

7 Conclusion

In this paper, we have proposed a new approach for the multi-period portfolio optimization problem using simulated sample paths directly and examined the efficacy of our proposed model through some numerical experiments. Since there is no need to convert a set of simulated sample paths into a scenario tree, which often leads to significant methodological and computational difficulties, it has become possible to obtain optimal solutions quickly through our proposed model. We combine two different types of risk chance constraints and CVaR risk management with a new framework of optimal decisions. Using CVaR risk measurement, the model can be solved with linear programming techniques. The stability of the optimal solutions may be improved by imposing additional risk chance constraints which can be

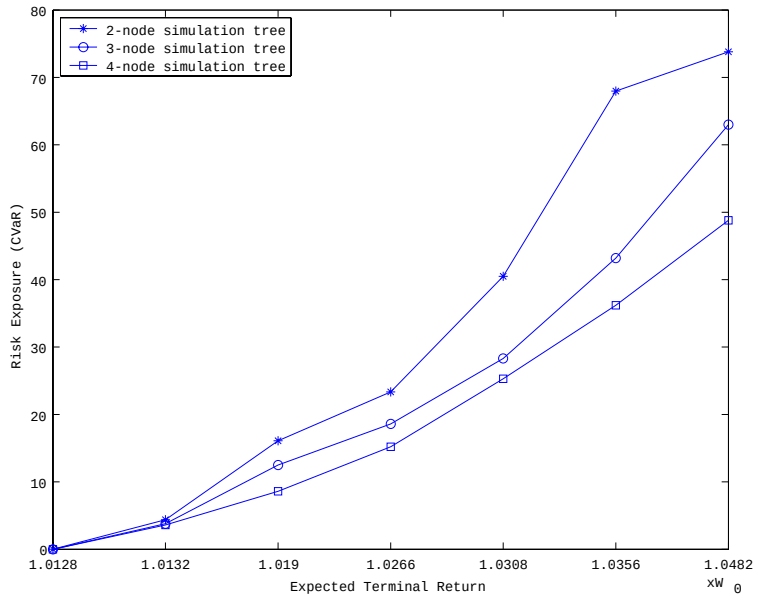

Figure 8: Efficient Frontiers for n-node simulation tree ($I = 1000$ sample paths).

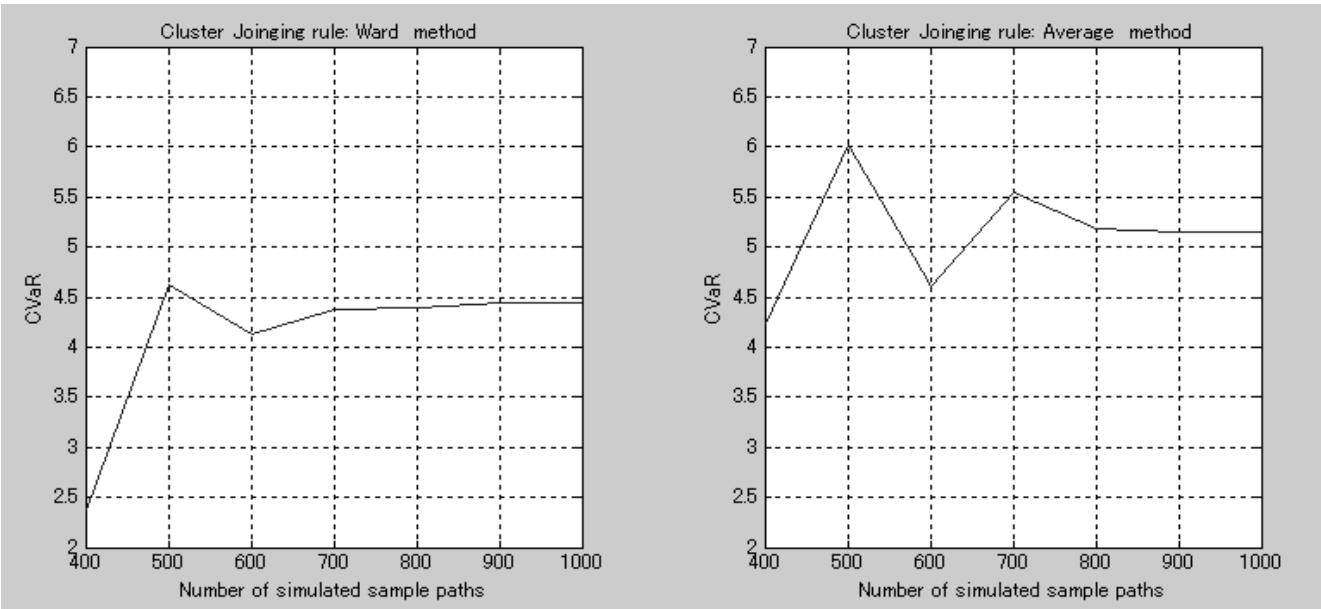

Figure 9: Convergence of different clustering methods ($W_E = 1.0132 \times W_0$).

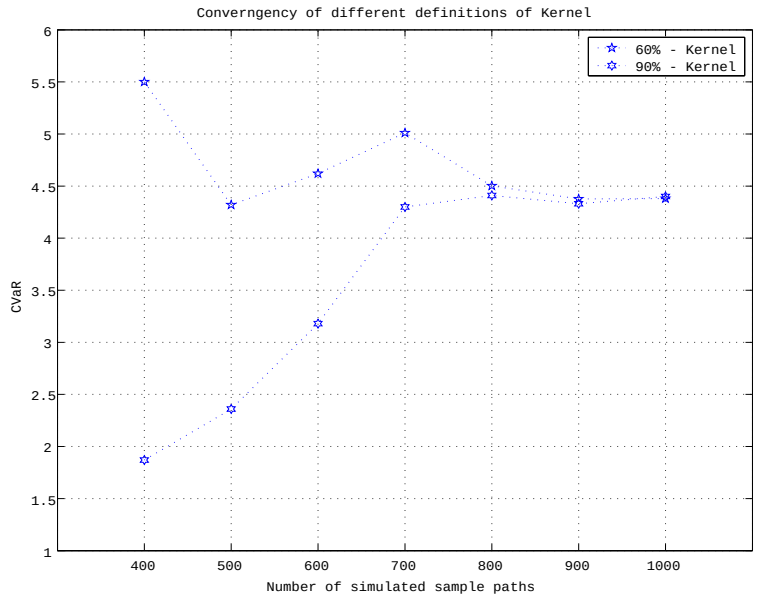

Figure 10: Convergence of different definitions for Kernel. (Clustering method: Ward, $W_E = 1.0132 \times W_0$).

converted into SOCs.

Our approach adds flexibility to the decisions while using only a sample path representation of uncertainties, allowing us to avoid the explosion of the problem size required by a traditional stochastic programming approach. The flexibility comes from grouping at each decision making time a set of sample paths that exhibit similar characteristics, and from restricting decisions to vary among different groups of sample paths. We obtain truly dynamic decisions at moderate computational expense, while allowing for extensive uncertainty through the use of paths.

Clearly, the choice of grouping method of simulated sample paths will affects the obtained optimal solution. Our experiments indicate that due to the additional risk chance constraints, two commonly used grouping methods tend to yield almost the same optimal value when the number of sample paths increases. From the computational point of view, taking advantage of such a property, we can reduce the problem size and required computational time. But further research should still have to be done to verify such a property when applying other grouping methods, especially a specific user designed grouping method.

References

[1] C. Acerbi and D. Tasche. On the coherence of expected shortfall. *Journal of Banking and Finance*, 26 (7):1487–1503, 2002.

[2] P. Artzner and P. Delbaen. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 1999.

[3] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[4] E.J. Bijnen. Cluster analysis: Survey and evaluation of techniques. *Tilburg University Press*, 1973.

[5] P. Boyle, M. Broadie, and P. Glasserman. Monte carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21:1267–1321, 1997.

[6] S.P. Bradley and D.B. Crane. A dynamic model for bond portfolio management. *Management Science*, 19:139–151, 1972.

[7] Andrew J.G. Cairns, David Blake, and Kevin Dowd. Optimal dynamic asset allocation for defined-contribution pension plans. *the 10th AFIR Colloquium, Tromsoe*, June 2000. the 10th AFIR Colloquium, Tromsoe.

[8] G.B. Dantizig and A. Mandansky. On the solution of two-stage linear programs under uncertainty. *the 4th Berkely Symposium on Mathematical Statistics and Probability*, 165-176, 1961. the 4th Berkely Symposium on Mathematical Statistics and Probability.

[9] Carino D.R., T. Kent, D.H. Myers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, and W.T. Ziemba. The russel-yasuda kasai model: An asset/ liability model for a japanese insurance company using multistage stochastic programming. *Interfaces*, 24 (1):29–49, 1994.

[10] M. Fukushima. *Fundamentals of Non-Linear Optimization*. Asakura-Shoten, 2001.

[11] N. Hibiki. Multi-period stochastic programming models for dynamic asset allocation. *the 31st ISCIE International Symposium on Stochastic Systems theory and Its Applications*, pages 37–42, 2000.

[12] N. Hibiki. Multi-period stochastic programming models using simulated paths for strategic asset allocation. *Journal of Operations Research Society of Japan*, 44:169–193, 2001.

[13] R.S. Hiller and J. Eckstein. Stochastic dedication: Designing fixed income portfolios using massively parallel benders decomposition. *Management Science*, 39:1422–1438, 1993.

[14] C. Joy, P. Boyle, and K.S. Tan. Quasi-monte carlo methods in numerical finance. *Working paper, University of Waterloo, Ontario, Canada N2L 3G1*, 1995.

[15] M.I. Kusy and W.T. Ziemba. A bank asset and liability management model. *Operation Research*, 34:356–376, 1986.

[16] M. S. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs and bounds on risk. *Working paper, Stanford University*, 1999. (Submitted to Operations Research, Oct 2002).

[17] R. Merton. *Continuous-Time Finance*. Revised edition, (Oxford, UK: Basil Blackwell), 1993.

[18] M.A. Milevsky. Optimal asset allocation towards the end of the life cycle: To annuitize or not to annuitize? *The Journal of Risk and Insurance*, 65(3):401–426, 1998.

[19] J.L. Moran-Lopez and J.M. Sanchez. *Theory and Applications of the Cluster Variation and Path Probability Methods*. Plenum Press, 1996.

[20] G. Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. *in S. Uryasev, ed., Probabilistic Constrained Optimization: Methodology and Applications, Kluwer Academic Publishers*, 2000.

[21] R.T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2 (3):21–41, 2000.

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

[23] R. Van Slyke and R.JB. Wets. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematicz*, 17:638–663, 1969.

[24] J.F. Sturm. *Using SeDuMi 1.0x, A Matlab Toolbox for Optimization Over Symmetric Cones*. Available at http://www.unimass.nl/ sturm, 1998.

[25] J.F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[26] Stan Uryasev. *Conditional Value-at-Risk (CVaR): Algorithms and Applications*. University of Florida, 2001.

[27] P. Vanini and L. Vignola. *VaR, CVaR and Mean-Downside Risk Portfolio Selection*. (Preliminary Draft), 2001.

[28] R.J-B. Wets. *Stochastic Programming.* in G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd, eds, Optimization, Handbooks in Operations Research and Management Science; Vol. 1, North-Holland, Amsterdam, 1990.

[29] S.A. Zenios. High performance computing for financial planning: The last ten years and the next. *Parallel Computing*, 25:2145–2175, 1999.

[30] T.W. Ziemba, M.J. Mulvey, and et al. *Worldwide Asset and Liability Modelling.* Cambridge University Press, 1998.

Appendix

A Compact Formulation

Although the original formulation is easy to understand, a compromise still has to be made between describing the precision of the underlying uncertainty and making conditional decisions with relatively high flexibility. Taking a closer look at the constraints of the formulation, we find that the number of decision variables $z(ß_t)$ represent to the investment vector depends only on the number of bundles in the decision tree, while the number of decision variables $v_t^{(i)}$ represent to the investment quantities in riskless assets depends on both the number of simulated sample paths and the number of periods. Therefore, by treating risk and riskless assets separately, the number of decision variables can be decreased substantially, thereby the problem size and the computational time can also be reduced. Another reason for treating the risky assets and riskless assets separately is that their totally different instinct properties. That is, the return rate for a riskless asset is already determined at the beginning of each period; while the return rate for a risk asset (such as stock, bond, etc.), remains unknown until the end of that period.

The equality constraints (10)-(13) in the formulation may be rewritten as inequality constraints by eliminating the cash variables. However, this does not mean that the cash variables are excluded from the asset allocation decision. Instead, cash variables emerge implicitly in the resulting compact formulation and can be calculated using risk asset variables.

A.1 Elimination of Cash Variable v_0 at the Beginning of the Planning Horizon

We can rewrite the cash variable v_0 as follows:

$$v_0 = -P_0' z(ß_0) + W_0, \tag{75}$$

and derive the following inequality constraint using the non-negative condition of v_0,

$$P_0' z(ß_0) \leq W_0. \tag{76}$$

A.2 Elimination of Cash Variable $v_t^{(i)}$ along a Sample Path i at Time t

Along a path i at time $t = 1$, the cash variable $v_1^{(i)}$ can be represented as follows:

$$\begin{aligned} v_1^{(i)} &= -(P_1^{(i)})' z(ß_1) + (P_1^{(i)})' z(ß_0) + (1 + r_0) v_0 \\ &= -(P_1^{(i)})' z(ß_1) + (P_1^{(i)})' z(ß_0) + (1 + r_0)(W_0 - P_0' z(ß_0) \end{aligned} \tag{77}$$

$$
\begin{aligned}
&= -(P_1^{(i)})'z(ß_1) + (P_1^{(i)} - (1+r_0)P_0)'z(ß_0) + (1+r_0)W_0 \\
&= -(P_1^{(i)})'z(ß_1) + (\eta_1^{(i)} * P_0)'z(ß_0) + (1+r_0)W_0,
\end{aligned}
$$

and can be transformed to an equivalent inequality constraint as well by using the non-negative condition,

$$
(P_1^{(i)})'z(ß_1) - (\eta_1^{(i)} * P_0)'z(ß_0) \leq (1+r_0)W_0. \tag{78}
$$

In general,$v_t^{(i)}$ can be reformulated as:

$$
\begin{aligned}
v_t^{(i)} &= -(P_t^{(i)})'z(ß_t) + (\eta_t^{(i)} * P_{t-1}^{(i)})'z(ß_{t-1}) + (1+r_{t-1}^{(i)})(\eta_{t-1}^{(i)} * P_{t-2}^{(i)})'z(ß_{t-2}) \\
&+ \ldots + \prod_{k=1}^{t-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) + \prod_{k=0}^{t-1}(1+r_k^{(i)})W_0,
\end{aligned} \tag{79}
$$

and its equivalent inequality constraint is straightforward,

$$
\begin{aligned}
(P_t^{(i)})'z(ß_t) &- (\eta_t^{(i)} * P_{t-1}^{(i)})'z(ß_{t-1}) - (1+r_{t-1}^{(i)})(\eta_{t-1}^{(i)} * P_{t-2}^{(i)})'z(ß_{t-2}) \\
&- \ldots - \prod_{k=1}^{t-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) \leq \prod_{k=0}^{t-1}(1+r_k^{(i)})W_0,
\end{aligned} \tag{80}
$$

where

$$
\begin{aligned}
\eta_{jt}^{(i)} &= p_{jt}^{(i)} - (1+r_{t-1}^{(i)})p_{jt-1}^{(i)} \\
&= (1+\mu_{jt}^{(i)})p_{jt-1}^{(i)} - (1+r_{t-1}^{(i)})p_{jt-1}^{(i)} \\
&= (\mu_{jt}^{(i)} - r_{t-1}^{(i)})p_{jt-1}^{(i)}.
\end{aligned} \tag{81}
$$

Assuming $\mu_{jt-1}^{(i)}$ as the return rate of unit asset j, we interpret $\eta_{jt-1}^{(i)}$ as the net (surplus) profit of asset j, compared to the interest rate r_{t-1}, because of the investment activity at the beginning of the period $[t-1, t]$.

Evidently, $v_t^{(i)}$ is the fraction left after investment on risky assets $(P_t^{(i)})'z(ß_t)$ from the total wealth $w_t^{(i)}$. Meanwhile, $w_t^{(i)}$ of path i at time t consists of two parts: (1) the cumulative amount of the initial wealth w_0 from the beginning point of the planning horizon till time t, $\prod_{k=0}^{t-1}(1+r_k^{(i)})W_0$; (2) sum of cumulative amount of risky investment profits generated at the end of each period until t, *i.e.*, $\prod_{k=1}^{t-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0)$ is the cumulative amount of first-period risky investment profit until time t.

A.3 Terminal Wealth $W_T^{(i)}$ of Sample Path (i)

From the explanation above, the terminal wealth $W_T^{(i)}$ of sample path i can be rewritten as follows:

$$
W_T^{(i)} = (\eta_T^{(i)} * P_{T-1}^{(i)})'z(ß_{T-1}) + (1+r_{T-1}^{(i)})(\eta_{T-1}^{(i)} * P_{T-2}^{(i)})'z(ß_{T-2})
$$

$$+ \quad \ldots + \prod_{k=1}^{T-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) + \prod_{k=0}^{T-1}(1+r_k^{(i)})W_0. \tag{82}$$

A.4 Compact Formulation

After elimination the cash variables, the original formulation can be represented as follows:

$$\text{Min} \quad \frac{1}{I}\sum_{i=1}^{I} q^{(i)} \tag{83}$$

$$\text{s.t.} \quad P_0' z(ß_0) \le W_0; \tag{84}$$

$$(P_1^{(i)})'z(ß_1) - (\eta_1^{(i)} * P_0)'z(ß_0) \le (1+r_0)W_0, \tag{85}$$
$$(i \in V(ß_1), ß_1 \in B_1);$$

$$(P_t^{(i)})'z(ß_t) - (\eta_t^{(i)} * P_{t-1}^{(i)})'z(ß_{t-1}) - (1+r_{t-1}^{(i)})(\eta_{t-1}^{(i)} * P_{t-2}^{(i)})'z(ß_{t-2}) - \ldots$$
$$- \prod_{k=1}^{t-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) \le \prod_{k=0}^{t-1}(1+r_k^{(i)})W_0, \tag{86}$$
$$(i \in V(ß_1)\bigcap \ldots \bigcap V(ß_k), ß_k \in B_k, k = 0, \ldots, t; t = 2, \ldots, T-1);$$

$$-q^{(i)} - (\eta_T^{(i)} * P_{T-1}^{(i)})'z(ß_{T-1}) - (1+r_{T-1}^{(i)})(\eta_{T-1}^{(i)} * P_{T-2}^{(i)})'z(ß_{T-2}) - \ldots$$
$$- \prod_{k=1}^{T-1}(1+r_k^{(i)})(\eta_1^{(i)} * P_0)'z(ß_0) \le \prod_{k=0}^{T-1}(1+r_k^{(i)})W_0 - W_G, \tag{87}$$
$$(i = 1, \ldots, \boldsymbol{I}; i \in V(ß_k), ß_k \in B_k, k = 0, \ldots, T-1);$$

$$\frac{1}{I}\sum_{i=1}^{I} W_T^{(i)} \ge W_E; \tag{88}$$

$$z(ß_t) \ge 0, (t = 0, \ldots, T-1; s \in S_t); \tag{89}$$

$$q^{(i)} \ge 0, (i = 1, \ldots, I).$$

Although the problem size will grow quickly if we increase the number of simulation paths in order to obtain a detailed description of the underlying uncertainty, the coefficient matrix of the problem remains to be sparse (see Figure 4). Many advanced mathematical programming software packages such as SeDuMi [25] can handle large-scale optimization problem efficiently by exploiting sparsity of the coefficient matrix.

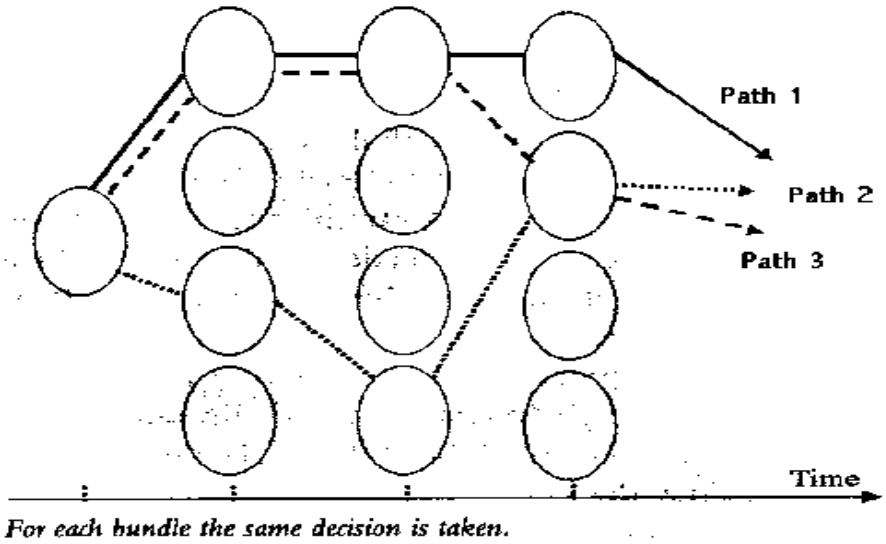

Figure 11: Grid-Net Modelling Description

B Gird-Net Modelling Description

At time t, sample paths are grouped in several bundles, $t = 1, \ldots, T-1$. Paths that are in two different groups at time t can pass through the same group at either earlier or later time(or both). Also, paths that pass through one node at time t do not necessarily pass through the same node at any other time. See figure (11) for the setup of Gird-Net.

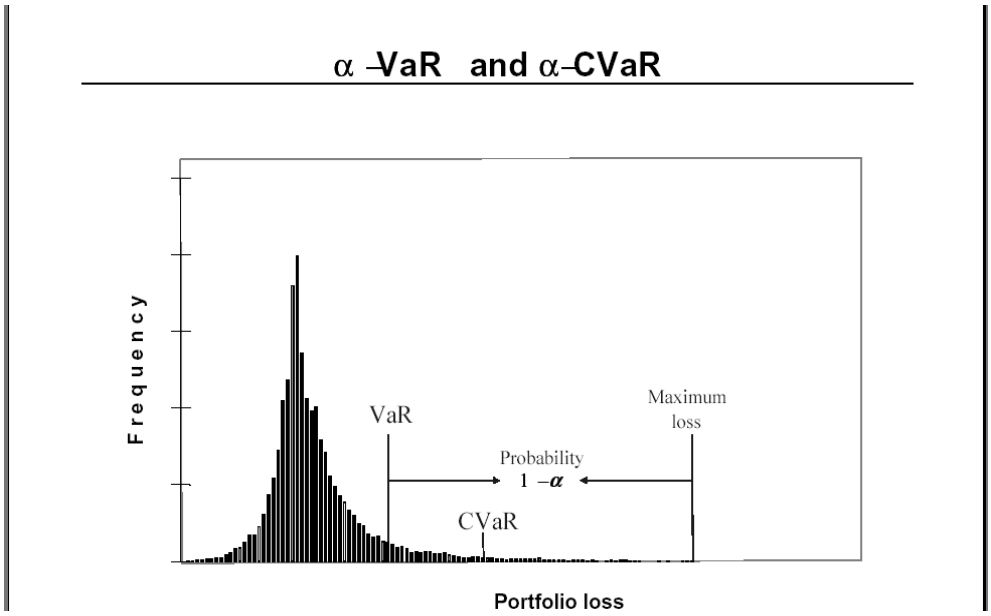

Figure 12: VaR and CVaR

C Illustration of VaR and CVaR