Endowment Fund Management Using a Stochastic Programming Approach

Michelle Zhang and Hujia Yu
Department of Management Science & Engineering
Stanford University
{mzhang8, hujiay}@stanford.edu

Abstract

This paper explores designing and implementing a system for the financial planning of a simplified endowment fund considering the following four asset classes: stocks, bonds, alternative investments, and cash. We formulate a multi-stage, specifically a three-stage, stochastic programming approach with a planning horizon of three years. In addition, we model yearly returns of our assets with a lognormal multivariate distribution fitted on historical data and incorporate state-dependent asset transaction costs. Using GAMS, we solve our system using various utility functions and analyze the results in order to determine the optimal asset allocation and trade-offs between payouts and long-term probability of loss of capital. We also try a CVaR formulation as an alternative way of measuring downside risk. From our in-sample and out-of-sample testing results, we conclude that when solving our model with the power utility function, we can meet a payout ratio of 4% beginning in the first year, whereas either CVaR or the piecewise linear utility function cannot do so until the second year. Furthermore, choosing the CVaR minimization method gives higher returns for the second and third year comparatively.

1 Introduction

Endowment funds, often held by universities, nonprofits, and other large organizations, involve investing a majority of acquired capital in a portfolio in addition to maintaining a regular payout schedule funded from the investment returns. Unlike classic portfolio optimization, endowment portfolio optimization places a heavier emphasis on the risk characteristics of its assets, including those that are not as liquid due to the nature of the fund. Merton (1993) describes the tendency for there to be a lack of alignment between the objective function for measuring the optimal portfolio and the ultimate goals of the endowment itself, where traditional objectives such as mean-variance efficiency may not be as helpful. One of the major issues is the reduced liquidity of alternative investments, in which it becomes more difficult to cash out the asset during times of financial distress.

This paper approaches the endowment portfolio optimization problem by extending the traditional portfolio optimization model to that of a dynamic stochastic model that optimizes for a desired target wealth with some proportion of payout at the end of each period while considering the potential downside risk of the portfolio investments. We apply this model to a small set of assets meant to represent our four asset classes and test out various utility functions to find the most suitable for the risk preferences of our endowment portfolio.

2 Background

2.1 Stanford Endowment Fund

University endowments such as Stanford's are enormous investment pools from which many universities attain their financial support. As of August 2016, Stanford's endowment is valued at over $20 billion [6] and is used for providing the funds for financial aid and academic programs alike. Stanford's endowment contains over 8,000 funds, many of which are set aside for reasons decided by their donors. The return from the invested capital in the endowment contributes to the university's operating expenses, while the remaining return is reinvested for maintenance.

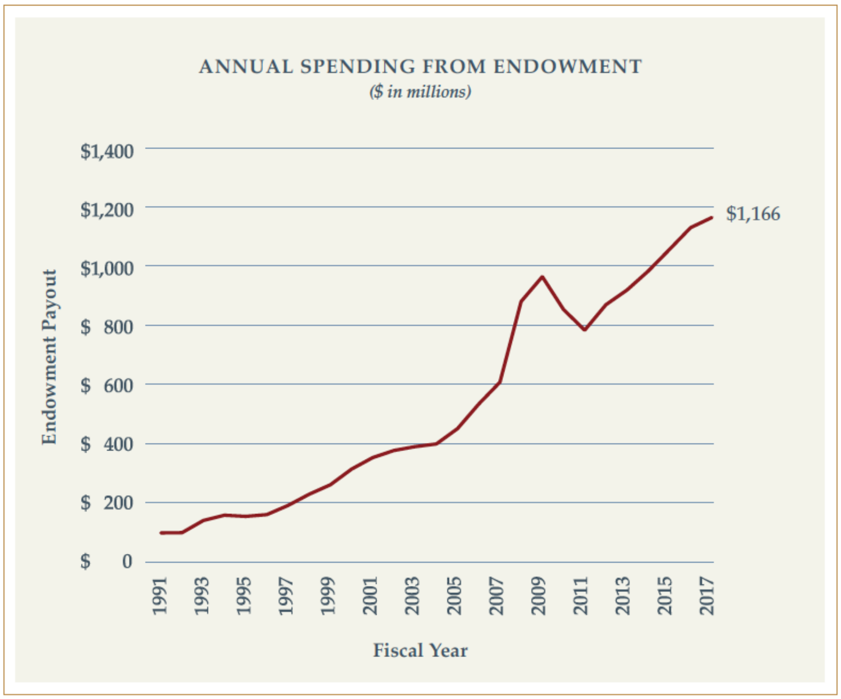

Figure 1: Annual endowment spending, taken from the Stanford University Investment Report.

From Figure 1, we can see that the amount of spending Stanford has done through its endowment has been increasing over time (except for the brief decline around the time of the 2008 financial crisis). Given an endowment size of about $22.4 billion, we can estimate that about 5.2% of the endowment was taken as payout for spending in 2016. Generally, an endowment may take somewhere around a fixed 4-5% payout cut. Several endowments such as the Yale endowment alternatively adjust the payout rate based on the performance of the portfolio.

Stanford's investment report additionally provides some insight into the Stanford Management Company's investment strategy for the endowment. Namely, we note that 25% of investments are in private equity [6], along with a noticeable proportion of capital in real estate and natural resources as well. When examining the 10-year returns of the endowment's asset classes, it appears that Stanford's investments in private equity and international equity have done particularly well above the benchmark amounts. We can hence see the importance of including a variety of asset classes in our portfolio, especially alternative investments, and understanding how such assets may influence the optimal allocation or outcome of the situation.

2.2 Related Work

Work done by Dantzig and Infanger in modeling a stochastic linear program for optimizing a multi-period portfolio [1] provides the baseline for which we extend our own model. Similar to our project, they consider some number n of risky assets with cash available to a decision maker with some initial wealth. At each point in time, the decision maker must decide whether and how to adjust his portfolio in order to optimize for returns at the end of the financial planning horizon. Assets are held, sold, or bought at each period according to some transaction cost for buying and selling each asset and appropriate constraints. We can then analyze the terminal value of the portfolio after all the periods.

Dantzig and Infanger's model does not allow for shortselling of assets or cash borrowing. Amount of each asset held, bought, and sold must then all be nonnegative. Many of these assumptions and simplifications we also take into account when designing our own model to optimize our endowment portfolio.

3 Approach

3.1 Overview

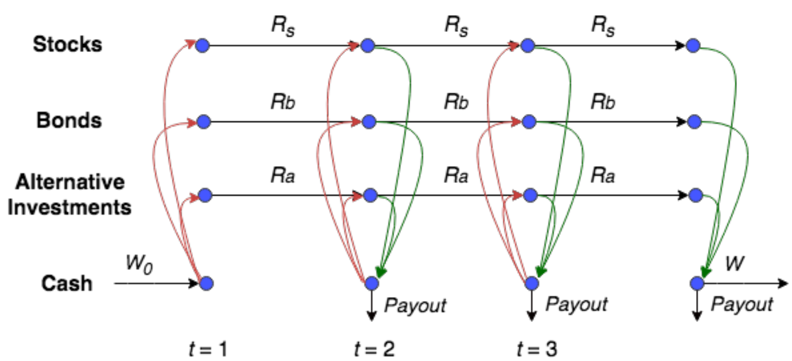

Figure 2: Diagram demonstrating flow of a multi-period investment over time.

Optimizing our endowment portfolio is ultimately a variation of the multi-period asset allocation problem with the addition of some payout requirement to our existing constraints. We thus solve our problem by formulating a multi-stage stochastic program with three stages.

Specifically, we have three time periods, $t = 1, \ldots, 3$ with representative assets $i, \ldots, n$ from our asset classes. Each time period has a length of one year. We represent cash as the asset indexed at $n + 1$. The first stage is deterministic, resulting an initial allocation of our assets. From there, we have two stochastic stages where we, the investor, have the ability to rebalance the portfolio based on the most recent returns and current value of our holdings. Each stage after the first furthermore takes out some amount of money as payout in order to mimic the behavior of an endowment.

At the end of the last stage, we then observe the terminal wealth of our portfolio. This formulation can naturally be increased to some arbitrary number of time steps, T. However, due to the uncertainty inherent in stochastic programming techniques, each additional stage increases the number of scenarios generated by an exponential factor. For example, having 20 samples in each stage results in $20 \times 20 \times 20 = 8{,}000$ total scenarios! This creates a significant burden in terms of computational cost, and so we restrict our model to three stages for the time being.

3.2 Model Formulation

With our n assets and cash, we denote x_i^t, y_i^t, z_i^t to be the dollar amounts held, sold, and bought for asset i at time t respectively. Prior to the first time period, we have only cash and zero holdings in our assets. Therefore our initial wealth can equivalently be expressed as $W_0 = R_{n+1}^0 x_{n+1}^0$. Since cash cannot actually be bought or sold like a true asset, we must set a constraint on the y_{n+1}^t and z_{n+1}^t to be zero.

The investor or decision maker can choose to hold onto, buy more, or sell some part of each asset when given the opportunity to rebalance the portfolio. When selling our asset, we decrease the value we have in this asset and add the dollars we received from the sale to our cash. Our cash funds can simultaneously be used to buy more amounts of other assets. Buying or selling assets unfortunately incurs some cost in reality. Each asset has an associated transaction cost per unit when bought or sold, which we denote to be ν_i^t and μ_i^t for asset i at time t. From these specifications, it follows that we must maintain the same total value of the portfolio after returns after rebalancing, regardless of the rearrangement of asset allocation.

State-dependent selling and buying transaction costs, μ_i^t and ν_i^t, for each asset i can be modeled as $\mu_i^t = \beta_i + \beta_i(-\frac{r_i^t}{\bar{r}_i^t} + 1)$ and $\nu_i^t = \beta_i + \beta_i(\frac{r_i^t}{\bar{r}_i^t} - 1)$ for some baseline cost β_i for all risky assets $i = 1, \ldots, n$. This reasonably captures the behavior we want, as when a particular asset's return is lower than average for period t, the cost to sell increases, while the cost to buy decreases. The opposite relationship occurs when a particular asset's return is higher than average.

After allocating or reallocating the endowment funds into our assets, x_i^t for $i = 1, \ldots, n+1$ remains constant until the beginning of the next stage, where we can observe the returns of our assets for the period. For an asset i that has some, say yearly, return r_i^t, the updated value of holdings in the asset becomes $R_i^t x_i^t$ for $R_i^t = 1 + r_i^t$. Recall that these returns are random and not known until the end of the time period. We let Ω to be the set of all possible outcomes of returns.

Finally, the payout can be thought of as a supplementary cash flow at the end of each period, defined by c^t. Note that this flow could also be positive, which would imply a situation where we receive some free influx of cash as opposed to paying out cash.

We seek to maximize an objective described by a utility function U of choice based on the total wealth, $W_{terminal}$, of the endowment portfolio at the end of the final stage. This leads us to the explicit formulation of our model:

$$Z = \max E[U(W_{terminal})] = \frac{1}{|\Omega|} \sum_{\omega_1, \omega_2, \omega_3} U(W_{terminal})$$

$$\begin{aligned}
&t = 1, \ldots, 3, \quad i = 1, \ldots, n+1 \quad R_{n+1}^0 x_{n+1}^0 \text{ given:} \\
&x_i^t + y_i^t - z_i^t = R_i^{t-1} x_i^{t-1}, \quad i = 1, \ldots, n \\
&x_{n+1}^{t-1} - \sum_{i=1}^{n} (1 - \mu_i^t) y_i^t + \sum_{i=1}^{n} (1 + \nu_i^t) z_i^t = R_{n+1}^{t-1} x_{n+1}^{t-1} + c^t, \\
&\quad x_i^t, y_i^t, z_i^t \geq 0, \quad i = 1, \ldots, n, \quad t = 1, \ldots, 3 \\
&\quad x_{n+1}^t \geq 0, \quad i = n+1 \quad t = 1, \ldots, 3 \\
&y_{n+1}^t, z_{n+1}^t = 0.
\end{aligned}$$

3.3 Utility Functions

We studied and experimented with several candidates for our model's utility function. Presuming some desired target wealth, W_{target}, the baseline model utilizes a piecewise linear function as the utility function:

$$U(W_{terminal}) = u \cdot \mathbb{1}_{W_{terminal} \geq W_{target}} - d \cdot \mathbb{1}_{W_{terminal} < W_{target}}, \quad d > u \geq 0.$$

In other words, this utility function has two components that encapsulate the upside slope and downside slope. Since we are risk-averse and are more concerned with the loss of money, there is greater weight placed on the downside slope using our parameters d and u so that a poor-performing portfolio is penalized more.

Our second utility function was the power utility function represented as

$$U(W_{terminal}) = \frac{W_{terminal}^{1-\gamma} - 1}{1 - \gamma}, \quad \gamma > 1$$

for some chosen risk aversion parameter γ. This utility function is also known as the isoelastic function and is a constant relative risk aversion (CRRA) utility function. The implication is that the fraction of wealth invested in our assets remains constant and is not dependent on the value of $W_{terminal}$ itself [3].

A variant of the quadratic utility function, a quadratic downside risk utility function,

$$U(W_{terminal}) = W_{terminal} - \frac{\lambda}{2}(\max(0, W_{target} - W_{terminal}))^2$$

for parameter λ was also tested. This utility function seemed to be an interesting option, applicable in the case where investors may be particularly averse to the downside risk of the terminal wealth being below the target wealth [3]. However, it is important to note that quadratic utility functions tend to assume that an investor would reduce the amount they invest in risky assets as wealth increases and as a result may not be the best choice to measure utility.

3.4 CVaR Minimization

An alternative approach to using a utility function as a risk measure is to simply minimize the conditional value-at-risk, CVaR, of the terminal wealth. CVaR, also called mean excess loss, mean shortfall, or tail VaR, is considered to be a more consistent and conservative measure of risk than VaR. We used a technique introduced by Rockafellar and Uryasev (2000), which calculates VaR and optimize CVaR simultaneously and applied this technique to our multi-stage stochastic programming problem. We minimize CVaR based on the returns in the final stage.

Let X be a random variable representing the value of a position. The value-at-risk is defined to be

$$VaR_\alpha(X) = \min\{V : P(-x > v) \leq \alpha\}$$

where α is loss tolerance. $VaR_\alpha(X)$ gives the smallest number V such that the probability of a loss greater than V is no more than α.

CVaR is the conditional expected value up to the α quantile,

$$CVaR_\alpha(X) = E\{-X | X \leq -VaR_\alpha(X)\}.$$

Since $CVaR_\alpha(X)$ is the average loss in the tail area up to the V calculated in $VaR_\alpha(X)$, it follows from definition that

$$CVaR_\alpha(X) \geq VaR_\alpha(X)$$

We can now formulate our problem as a linear programming CVaR minimization problem:

$$\begin{aligned} \min\ & V + \frac{1}{\alpha}\frac{1}{T}\sum_{t=1}^{T}\{v_t\} = CVaR_\alpha \\ \text{subject to}\ & e^\top \omega = 1 \\ & \bar{r}^\top \omega \geq \bar{r_p} \\ & v_t + \omega^\top r_t + V \geq 0 \\ & v_t \geq 0, \end{aligned}$$

where T is the total number of asset classes and ω is the vector of decision variables to be determined. Here r is a random vector representing asset returns, $X(\omega, r)$ is the value function of the portfolio, $X(\omega, r) = r_t^\top \omega$, and V is VaR_α.

We want to minimize v_t, the deviation from VaR_α when minimizing $CVaR_\alpha$. In our three-stage problem, $\omega^\top r_t$ from the final stage is plugged in for calculation. Since we already take into consideration the payout ratio at the end of each stage, $\bar{r}_p$ is set to zero most of the time.

To frame our model as a CVaR minimization problem, we add the above constraints to our existing multi-stage stochastic programming problem and change the objective function to minimize CVaR.

3.5 Estimating and Sampling Returns

At each stage of our model, the year's returns must be randomly sampled from a distribution to produce the various scenarios and outcomes when solving. We assume that the yearly returns of the assets are lognormally distributed, correlated with each other, and fit a multivariate distribution accordingly to our historical data.

Most of the data for the past ten years was readily available via Yahoo Finance and the `quantmod` package in R. For stock, bonds, alternative investments, and cash, we used one or two representative index funds for each: the S&P 500 Index (`^GSPC`), a collection of 500 large companies with common stock on the NYSE, for stock, the iShares IBoxx $ Investment Grade Corporate Bond ETF (`LQD`) for corporate bonds, the iShares 7-10 Year Treasury Bond ETF (`IEF`) for government bonds, and the SPDR Barclays Capital 1-3 Month Treasury Bill ETF (`BIL`) for cash. Alternative investments were represented solely by a hedge fund index taken from Eurekahedge.

Below, Figure 3 shows the returns of the various assets over the past ten years by month, beginning in February 2008.

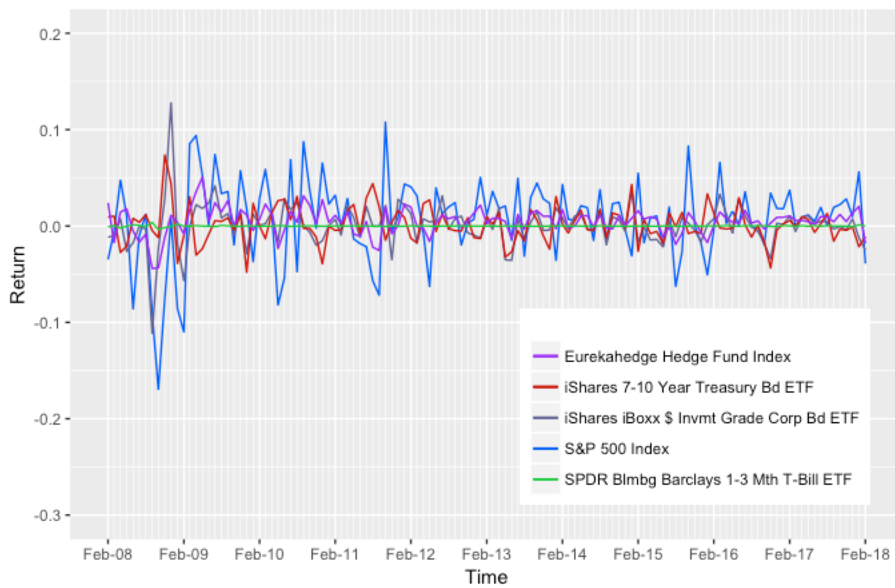

Figure 3: 10-year return for assets.

Once we retrieved the monthly returns for the assets, each return r_i was converted to the log return by applying the transformation $\log(1+r_i)$. We know that these new log random variables are normally distributed by definition, so we estimate the vector of means $\vec{\mu}$ and covariance matrix Σ of our log returns. The parameters $\vec{\mu}$ and Σ for our multivariate Gaussian are then scaled be in terms of yearly returns and are outputted to our model in GAMS.

Next, to sample the yearly returns for each stage, we use the following idea:

1. Begin with a vector of zeros $\vec{v} \in \mathbb{R}^n$.
2. For all $i = 1, \ldots, n$, draw a sample from the standard normal distribution $v_i \sim N(0, 1)$.
3. Calculate the Cholesky factorization, $\Sigma = LL^\top$.
4. Then, the random sample from the multivariate Gaussian is $L\vec{v} + \vec{\mu}$.

This procedure can be repeated as many times as we want to generate the number of returns needed for the model. For efficiency purposes, the Cholesky factorization of the covariance matrix is calculated once before any sampling from the fitted data.

4 Results

4.1 Parameters

For our baseline model, we decided on a fixed payout ratio of 4% of the initial wealth. Due to the potential issue of inaccurate estimates when pre-scaling wealth, the initial wealth is always set to 1 dollar, and any result can be appropriately adjusted to a larger scale, i.e. that of a \$1 million or \$1 billion endowment. In our case, this means that our model requires a payout of 0.04 dollars each year. When applicable, we used a wealth target of 1.5.

The baseline costs for the selling and buying transaction costs of an asset i were set to be the same. However, these costs were decided based on asset class. We roughly estimated that transaction cost for stocks, bonds, alternative investments, and cash should be 50, 25, 100, and 0 basis points (1/100th of a percent) respectively. Hence, shifting our holdings in the hedge fund index becomes the most expensive without the state-dependent factor.

When testing the different utility functions and versions of the model, the standard sample number was 30, which amounts to $30 \times 30 \times 30 = 27,000$ samples. This takes about a minute or so to run, but increasing to 50 samples took significantly longer time and did not seem to yield much difference in results.

For the utility functions, the parameters were set to $u = 1, d = 10$ for the piecewise linear utility function, $\alpha = 20$ for the power function, and $\lambda = 10$ for the quadratic variant function.

Fitting the historical data of monthly asset returns to a multivariate Gaussian resulted in the following mean and covariance matrix:

Instrument	**Mean Return**
S&P 500	0.06044
Corp. bond	-0.03083
Gov. bond	-0.00891
Hedge fund	0.054306
Cash	-0.00267

Table 1: Mean estimates.

	S&P 500	**Corp. bond**	**Gov. bond**	**Hedge fund**	**Cash**
S&P 500	0.023490	0.006461	-0.001131	0.000149	0.005937
Corp. bond	0.006461	0.021773	0.011162	0.000819	0.002413
Gov. bond	-0.001131	0.011162	0.0088908	0.000468	-0.002482
Hedge fund	0.000149	0.000819	0.000468	0.000543	0.000465
Cash	0.005937	0.002413	-0.002482	0.000465	0.002328

Table 2: Covariance matrix estimate.

It is clear that the S&P 500 and hedge fund index had the largest returns on average, while other asset returns were closer to zero. Note that these values are in terms of the log return and not the R_i returns used directly in our model.

4.2 Piecewise Linear Utility

Given optimal asset allocations for each of thirty scenarios, we can visualize the fraction of total wealth we have in each asset and additionally plot a histogram for the varying wealth totals for the second stage as in Figure 4.

We can see that in general, we would invest all of our funds into the stock asset at the second stage. Once wealth increases, we begin to take on more risk and invest in an increasing proportion of the hedge fund index as well. The majority of the total wealth in the thirty scenarios seems to lie around the 1.0 to 1.2 range. Nonetheless, there is a noticeable number of scenarios that had wealth below 1.0, which indicates some loss from our initial capital.

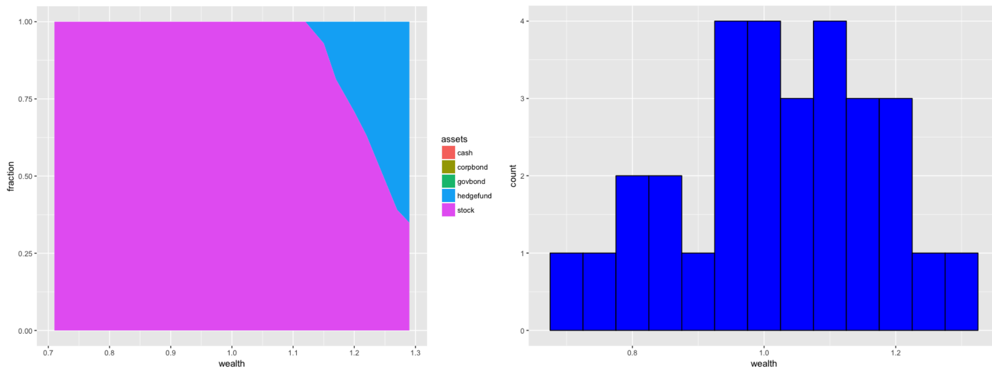

Figure 4: Stage 2 asset allocation and wealth.

In the third stage, we have a total of $30^2 = 900$ scenarios, which we can straightforwardly create the same visualization of the asset allocation and histogram plot of wealth totals for this subsequent stage as in Figure 5.

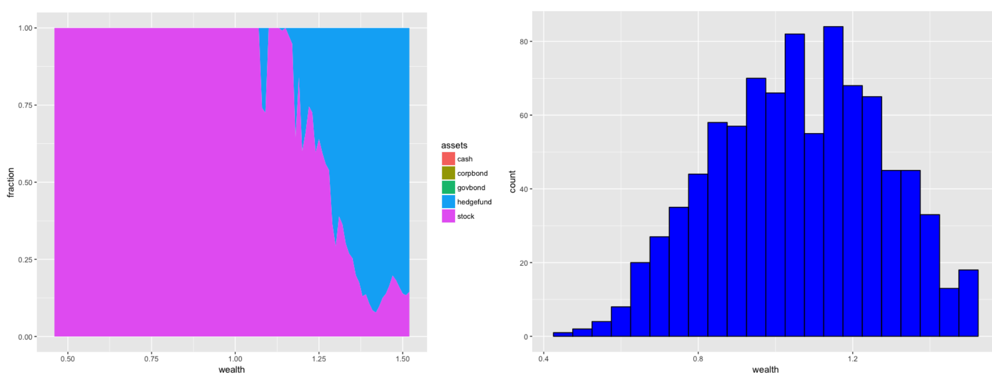

Figure 5: Stage 3 asset allocation and wealth.

The asset allocation looks more interesting compared to in stage two, but the optimal asset allocation for the portfolio still tends to rely heavily on investing everything in stock until a total wealth of at least 1.0 is reached. Interestingly, when our wealth approaches 1.5 in the third stage, we drastically change our position and invest most of our funds into the hedge fund index.

From the histogram plot, we observe that the distribution of the total wealth begins to look closer to normal but possibly very slightly skewed, and we hit above the initial wealth value more than half the time.

We briefly explored adjusting the fixed payout ratio and wealth target values with the piecewise linear utility function to ask how the asset allocation might change as a consequence. In general, if we doubled the payout ratio to 8% and increased the wealth target by 0.5 to 2.0, our asset allocation for the second stage would be purely dedicated to investing in stock, regardless of the possible wealth totals from the scenarios. On the other extreme, when we had no payout, we saw an increased and earlier investment in other assets, as in the hedge fund index.

4.3 Power Utility

We can repeat the same graphical analysis on the optimal asset allocations and wealth values generated by the second and third stages of the model using the power utility function, displayed in Figure 6.

As we expected, because this utility function is a CRRA utility function, the optimal asset allocation consistently holds the same proportion of assets, independent of wealth in both the second

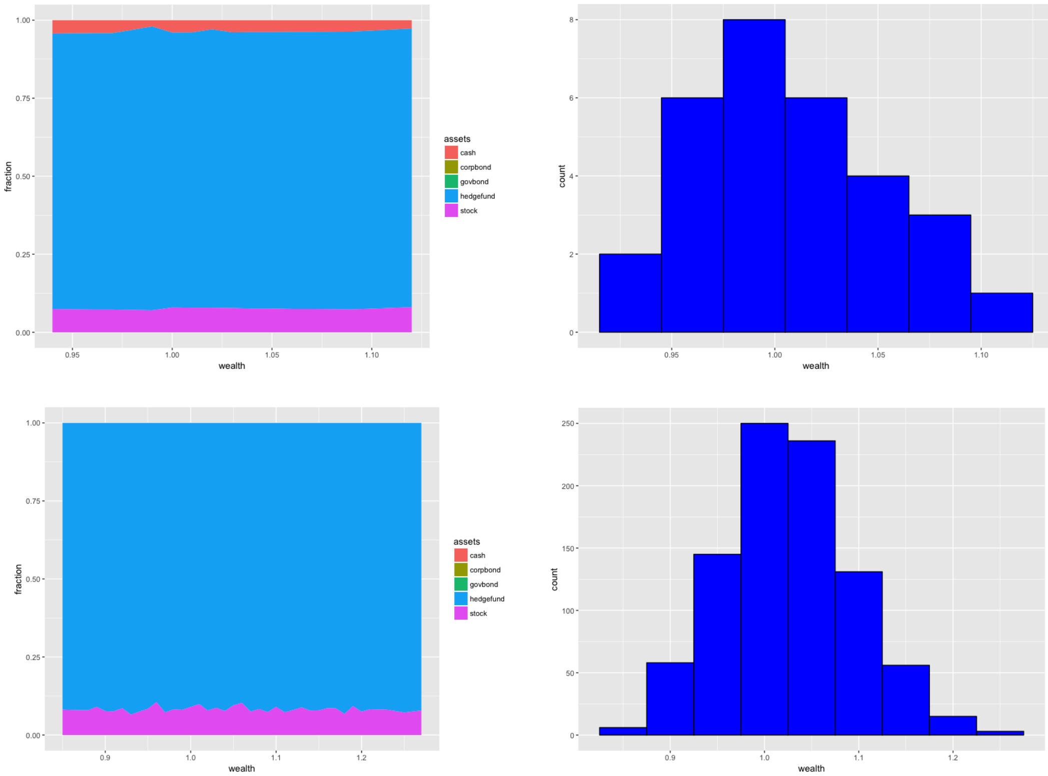

Figure 6: Stages 2 (top) and 3 (bottom) asset allocation and wealth.

and third stage. Unlike the results when utilizing the piecewise linear utility function, there is some fraction of asset allocation that goes to a third asset, cash, for the simulated scenarios. Similar to the previous section, however, a majority of the total wealth outcomes tends to be above the initial wealth, which translates to some positive return on our portfolio, in most of the scenarios.

There may be some margin of error in calculation that needs to be considered since the asset allocation graphs for the two stages did not have entirely identical outputs.

4.4 Quadratic Downside Risk Utility

We can again repeat the same graphical analysis on the optimal asset allocations and wealth values generated by the second and third stages of the model using the quadratic downside risk utility function, with output for the second stage displayed in Figure 7.

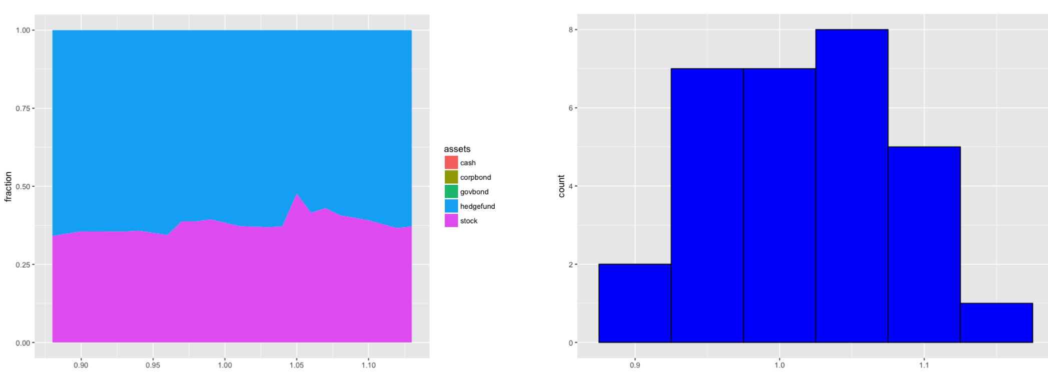

Figure 7: Stages 2 asset allocation and wealth.

The output for the third stage, not shown, gives about the same asset allocation for a wider range of wealth in the third stage. As the results in asset allocation do not vary much across different wealth values, we might believe that the parameter for the utility function should have been tuned better along with perhaps changing the wealth target to see how the results are affected, given that this particular utility function relies on the specified wealth target.

4.5 CVaR

Here we present the results from the GAMS solver for the CVaR minimization approach on our three-stage stochastic programming problem, where the decision variables in the final stage is plugged in the CVaR formulation.

α	**Sample Size**	**VaR**	**CVaR**	**Iterations**	**Time (sec)**
0.01	20	0.075	0.075	3584	2.68
0.01	30	0.066	0.066	12163	22.98
0.01	50	0.073	0.073	46565	327.51
0.05	20	0.043	0.075	5396	2.88
0.05	30	0.052	0.064	11860	21.03
0.05	50	0.052	0.063	158294	1595.91
0.1	20	0.041	0.058	9977	5.94
0.1	30	0.048	0.057	20357	28.70
0.1	50	0.042	0.054	187718	1833.82

Table 3: Results from the CVar approach.

First of all, as α increases, CVaR decreases. This is intuitive, since as we increase our risk tolerance, less of our portfolio value will be at that amount of risk. Also, risk tolerance essentially corresponds to a quantile in the risk distribution, so a smaller α indicates a larger VaR, and in turn a larger CVaR.

Moreover, we can observe from the table that the time it takes to solve the optimization problem increases exponentially as sample size increases, which is understandable given that the complexity of simulation is of the order $O(\text{sample size}^3)$. Setting sample size to 50 per stage, for example, increases the number of scenarios we have to $50^3 = 125{,}000$ from a mere $20^3 = 8{,}000$ with 20 samples.

The advantage, nevertheless, of having more scenarios is that we can get a more accurate measure of the return distribution, and arguably a better estimate of risks. This is especially since CVaR is measuring tail distribution, which is more difficult to be sampled from. A greater sample size then allows for more data points in the tail area and gives a better risk measure for that reason. This is shown in the CVaR column where for each α level, CVaR decreases as sample size increases, indicating a better-defined tail distribution.

Lastly, we can observe from the table that CVaR is always greater than or equal to VaR, which is to be expected, based on the definition of CVaR.

5 Testing

5.1 In-sample Testing

We solve the optimization problem first using Monte Carlo sampling. Then, we estimate the expected return of the single-period utility maximizing problem of each period. The in-sample testing is simply to evaluate the solution by getting the expected return at each stage of the solved model.

5.2 Out-of-sample Testing

When we have solved the optimization problem, we obtain a single, deterministic solution for stage one, s re-balancing scenarios for stage two, and s^2 re-balancing scenarios for stage three for a sample size s. We want to test this solution on new data and observe the performance.

During testing, instead of calculating expected return on existing sampled data, we again use Monte Carlo sampling to sample new data from the same distribution of returns derived previously. We then set our first stage portfolio weights to be the optimal first stage portfolio weights obtained from solving the optimization problem from initially. Next, we solve for the second and third stage scenarios, calculating expected return at the end of each stage.

The in-sample and out-of-sample testing results for each utility function and CVaR minimization model are given in the tables below.

Test	α	**Target Return**	**Payout**	**Sample Size**	**Stage 1**	**Stage 2**	**Stage 3**
In-sample	0.01	0.07	0.0	30	0.0687	0.1481	0.2054
Out-of-sample	0.01	0.07	0.0	30	0.0203	0.0961	0.1566

Table 4: CVaR testing.

Test	**Payout**	**Sample Size**	**Wealth Target**	**Stage 1**	**Stage 2**	**Stage 3**
In-sample	0.04	20	1.5	0.0685	0.102	0.1347
Out-of-sample	0.04	20	1.5	0.02	0.0518	0.0824

Table 5: Piecewise linear utility function testing.

Test	**Payout**	**Sample Size**	**Wealth Target**	**Stage 1**	**Stage 2**	**Stage 3**
In-sample	0.04	20	1.5	0.05	0.0692	0.0891
Out-of-sample	0.04	20	1.5	0.05	0.0501	0.0579

Table 6: Power utility function testing.

For our CVaR testing results, the out-of-sample testing based on first stage results occasionally renders infeasible results. To avoid these infeasible results, we restrict the model to using expected return and set the payout ratio in the formulation to be zero in both in-sample testing and out-of-sample testing. With expected return of 0.07 at the end of the third stage, out-of-sample testing results show that starting at the second stage, the model will be able to generate returns greater than 4% and will meet our payout ratio requirements. With expected return set to be 0.07, out-of-sample testing results on CVaR minimization model shows higher returns than those rendered by the other two utility functions.

The model using the piecewise linear utility function gives better results than that of the power utility function for both in-sample and out-of-sample tests. However, according to out-of-sample test results, returns generated in the former case can only meet the 4% payout requirement after the second stage, whereas the later can meet payout requirements from the very first stage.

6 Conclusion

While traditional portfolio optimization models may be well-suited for individual investors seeking to maximize returns, our paper focuses on solving the problem of optimizing an endowment portfolio, which places a higher emphasis on the risk of lost capital and demands a consistent payout schedule. We do this by formulating the problem as a multi-stage stochastic program with a time horizon of three years.

Through employing the utility functions defined, we find that with no diversification constraint, our simplified endowment fares optimally when we invest mostly in stock and alternative investments even with the higher-end transaction costs for the two asset classes. From the in-sample and out-of-sample testing results, we conclude that when our model is solved with the power utility function, we can always meet a payout ratio of 4%, whereas with either CVaR or the piecewise linear utility function, we can only meet payout ratio starting in the second year. Furthermore, CVaR minimization yields higher returns for the second and third years compared to the returns from the other two models.

6.1 Future Work

There are many extensions and improvements of this project to be investigated in more detail. For instance, it may be interesting to increase of number of assets for the portfolio as we recall that real endowments often an enormous diversification of different assets and asset classes. A turnover constraint could easily be added to cap the maximum amount that can be allocated into any one asset.

Results had a strong preference for the stock and alternative investments classes because our data appeared to have a rather low mean for bonds and cash, which could be possibly corrected in the future with more representative data.

Moreover, despite having tried several utility functions, the model can incorporate other, more complex utility functions depending on the circumstances of the investor. Payout was fixed for each time period in our implementation of the model but could very well be modeled as a distribution itself. State-dependent costs, too, could be refined to be even more realistic.

Our modeled returns for each successive period are not dependent on time or previous returns. In the future, it may be worthwhile to model the returns then using a vector autoregressive process and compare the results to our current optimal outcomes.

References

[1] Dantzig, G. B. & Infanger, G. (1993). Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research*, 45(1):59-76.

[2] Dantzig, G. B. & Wolfe, P. (1961). The Decomposition Algorithm for Linear Programs. *Econometrica*, 29(4):767-778.

[3] Infanger, G. (2006). Dynamic asset allocation using a stochastic dynamic programming approach. *Handbook of Asset and Liability Management*, volume 1, North Holland.

[4] Merton, R. C. (1993). Optimal Investment Strategies for University Endowment Funds. *Studies of Supply and Demand in Higher Education*, 211-242.

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

[6] Stanford Management Company. (2017). Stanford University Investment Report. Stanford, CA.