Stochastic Programming

Supervisor: Professor Vikas Vikram Singh

Vishavjeet Singh (2014MT60570)

Raunak Lohiya (2014MT60362)

Department of Mathematics

Indian Institute of Technology, Delhi

Acknowledgements

We would like to thank Prof Vikas Vikram Singh for guiding us throughout the project. His invaluable support and expertise greatly assisted us to complete our goals. During the first phase of the project, he also took weekly meetings to ensure that we understand the concepts clearly.
We would also like to thank Dr. Aparna Mehra for her guidance which gave our project a practical direction. This helped us to apply theoretical models in real life problems and precisely understand the challenges faced.

Abstract

The idea of the project is to understand what stochastic programming means and how it is used in real life. In the first part of our report, we present our learnings about different models including static models, 2-stage recourse models, chance constraint models and multi-stage models for inventory, multi-product assembly and portfolio selection.

In the second part, we implemented a lot of these models including static models for expectation maximization, variance minimization using different objective functions involving constraints like value at risk (VaR) and conditional value at risk (CVaR). Further, we implemented 2-stage recourse action problems for portfolio optimization and multi-product assembly. We also created multi-stage models using scenario tree generation with piecewise linear objective function and minimizing VaR with maximizing expectation and vice-versa.

Contents

Chapter 1

Introduction

Stochastic programming is an approach for modeling optimization problems that involve uncertainty. The probability distributions governing the data are known or can be estimated. The goal is to find some policy that is feasible for all the possible parameter realizations and optimizes the expectation of some function of the decisions and the random variables.

Two-stage programs are one of the most widely applied and studied stochastic programming models. Here the decision maker takes some action in the first stage, after which a random event occurs affecting the outcome of the first-stage decision. A recourse decision can then be made in the second stage that compensates for any bad effects that might have been experienced as a result of the first-stage decision. The optimal policy from such a model is a single first-stage decision and a collection of recourse decisions defining which second-stage action should be taken in response to each random outcome. Similarly multistage programming models can be formulated where one has to take the decisions at successive stages depending on the information available at the current stage.

Chapter 2

Theory

Some problems and formulations in stochastic programming from [1], [2].

2.1 Inventory Problems

Suppose that a company has to decide an order quantity x of a certain product to satisfy demand d. The cost of ordering is c $>$ 0 per unit. If the demand d is bigger than x, then a back order penalty of b $\geq$ 0 per unit is incurred. The cost of this is equal to b(d $-$ x) if d $>$ x, and is zero otherwise. On the other hand if d $<$ x, then a holding cost of h(x $-$ d) $\geq$ 0 is incurred. The total cost is then:

$$G(x, d) = cx + b[d - x]_+ + h[x - d]_+$$

We assume that $b > c$, i.e., the backorder cost is bigger than the ordering cost. The objective is to minimize the total cost $G(x, d)$ Therefore, if the demand is known, the corresponding optimization problem can be formulated in the form

$$\min_{x \geq 0} G(x, d)$$

Without non-negativity constraints, the problem reduces to a piecewise linear function with a minimum attained at $\bar{x} = d$:

$$\max\{(c - b)x + bd, (c + h)x - hd\}$$

Consider now the case when the ordering decision should be made before a realization of the demand becomes known. One possible way to proceed in such situation is to view the demand D as a random variable (denoted here by capital D in order to emphasize that it is now viewed as a random variable and to distinguish it from its particular realization d). We assume, further, that the probability distribution of D is known. This makes sense in situations where the ordering procedure repeats itself and the distribution of D can be estimated, say, from historical data.

Model 1

$$\min_{x \geq 0} \mathbb{E}[G(x, D)]$$

If the process repeats itself then, by the Law of Large Numbers, for a given (fixed) x, the average of the total cost, over many repetitions, will converge with probability one to the expectation $\mathbb{E}[G(x, D)]$.

The above problem gives a simple example of a recourse action. At the first stage, before a realization of the demand D is known, one has to make a decision about ordering quantity. At the second stage after demand D becomes known, it may happen that $d > x$. In that case the company can meet demand by taking the recourse action of ordering the required quantity dx at a penalty cost of $b > c$.
To obtain a closed form solution of the above problem, we consider the cumulative distribution function (cdf) $F(z) := Prob(D \leq z)$ of the random variable. Since the demand cannot be negative, $F(z) = 0$ for any $z < 0$. We could show that:

$$\mathbb{E}[G(x, D)] = b\mathbb{E}[D] + (c - b)x + (b + h) \int_0^x F(z)dz$$

Therefore, by taking the derivative, with respect to , and equating it to zero we obtain: $(b + h)F(x) + c - b = 0$

Hence an optimal solution is given by the quantile:

$$\bar{x} = F^{-1}(k)$$

where

$$\kappa := \frac{b - c}{b + h}$$

In real life, we obtain an empirical distribution using the history. It could be modelled as follows: Let the random variable D take values $d_1, ..., d_K$ (called scenarios) with respective probabilities $p_1, ..., p_K$. The cdf $F()$ is a step function with jumps of size p_k at each $d_k, k = 1, ..., K$.

$$\mathbb{E}[G(x, D)] = \sum_{k=1}^{K} p_k G(x, d_k)$$

Model 2

We could also consider solving for the worst case approach. Suppose for the moment that the random variable D has a finitely supported distribution, i.e., it takes values $d_1, ..., d_K$ (called scenarios) with respective probabilities $p_1, ..., p_K$. Our objective function becomes:

$$\min_{x \geq 0} \max_{d \in [l,u]} F(x, d)$$

$$\max_{d \in [l,u]} F(x, d) = \max[F(x, l), F(x, u)]$$

With the above equation, our minimization function becomes a convex function and optimal solution is obtained at:

$$x* = \frac{hl + bu}{h + b}$$

Thus, the way we form our objective function can lead to different kinds of solution.

Model 3

Chance Constraint

A lot of times, a person wants to have an upper bound on the cost function apart from minimizing it on average. Formally,

$$F(x, D) \leq \tau$$

for all possible realizations of D. This could be very restrictive if D takes large number of values. In these situations, it is useful to introduce the constraint that the probability of $F(x, D)$ being larger than is less than a specified value (significance level) $\alpha \in (0, 1)$. This leads to a chance (also called probabilistic) constraint as follows:

$$Pr\{F(x, D) > \tau\} \leq \alpha$$

or

$$Pr\{F(x, D) \leq \tau\} \geq 1 - \alpha$$

Using $F(x, D)$ from our inventory model, we obtain:

$$Pr\{F(x, D) \leq \tau\} = Pr\{\frac{(c + h)x - \tau}{h} \leq D \leq \frac{(b - c)x + \tau}{h}\}$$

Assuming H is the cdf of D,

$$Pr\{F(x,D) \leq \tau\} = H(\frac{(b-c)x+\tau}{h}) - H(\frac{(c+h)x-\tau}{h})$$

$$H(\frac{(b-c)x+\tau}{h}) - H(\frac{(c+h)x-\tau}{h}) \geq 1-\alpha$$

Model 4

Multistage Model

Let the company has a planning horizon of periods. Let be the random demand for $t = 1, ..., T$. Let y_t be the inventory level at $t = 1, ..., T$. Let the quantity ordered at t to replenish the inventory level be x_t.

The order quantity should be nonnegative therefore, $x_t \geq y_t$. Let d_t be the realization of D_t after the inventory is replenished, hence the next inventory level, at the beginning of period $t+1$ becomes $y_{t+1} = x_t - d_t$. Backlogging is allowed and y_t may become negative. The total cost incurred in period t is:

$$c_t(x_t - y_t) + b_t[d_t - x_t]_+ + h_t[x_t - d_t]_+$$

where c_t, b_t, h_t are the ordering ,backorder penalty and holding costs per unit at time t. We assume that $b_t > c_t > 0$ and $h_t \geq 0$ for $t = 1, ..., T$. Our objective is to minimize the expected value of the total cost over the planning horizon, i.e.,

$$min \quad \sum_{t=1}^{T} E\{c_t(x_t - y_t) + b_t[D_t - x_t] + h_t[x_t - D_t]_+\}$$

$$s.t. \quad x_t \geq y_t$$

$$y_{t+1} = x_t - D_t$$

$$t = 1, ..., T$$

Let $D_{[t]} = (D_1, ..., D_t)$ be the history of the demand process upto time t and $d_{[t]}$ be its particular realization. We assume that the conditional probability distribution of D_t, given $D_{[t-1]} = d_{[t-1]}$ is known. At stage T we need to solve the problem -

$$Min \quad c_T(x_T - y_T) + E\{b_T[D_T - x_T]_+ + h_T[x_T - D_T]_+ | D_{[T-1]} = d_{[T-1]}\}$$

$$s.t. \quad x_T \geq y_T$$

Let

$$Q_T(y_T, d_{[T-1]})$$

be the optimal value at stage T. For

$$t = T-1, ..., 2$$

the problem to be solved is-

$$Q_t(y_t, d_{[T-1]}) = min \quad c_t(x_t - y_t) + E\{b_t[D_t - x_t]_+ + h_t[x_t - D_t]_+ + Q_{t+1}(x_t - D_t, D_{[t]})|D_{[t-1]} = d_{[t-1]}\}$$

$$s.t. \quad x_t \geq y_t$$

Finally at the first stage we need to solve the problem

$$Min \quad c_1(x_1 - y_1) + E\{b_1[D_1 - x_1]_+ + h_1[x_1 - D_1]_+ Q_2(x_1 - D_1, D_1)\}$$

$$s.t. \quad x_1 \geq y_1$$

2.2 Multi Product Assembly

There is a manufacturer who produces n products. There are m different parts which have to be ordered from third-party suppliers. A unit of product i requires a_{ij} units of part j, where $i = 1, ..., n$ and $j = 1, ..., m$. The demand for the products can be modeled as a random vector $D = (D_1, ..., D_n)$. Before the demand is known, the manufacturer may preorder the parts from outside suppliers at a cost of c_j per unit of part j. After the demand D is observed, the manufacturer may decide which portion of the demand is to be satisfied, so that the available numbers of parts are not exceeded. It costs additionally l_i, to satisfy a unit of demand for product i, and the unit selling price of this product is q_i. The parts not used are assessed salvage values $s_j < c_j$. The unsatisfied demand is lost. The aim is to decide x the number of parts to be ordered before the demand is realized such that the expected production cost is minimized. After D is realized let the number of units produced be Z and number of parts left be Y. The problem can be modeled as a **two stage problem** -
First stage -

$$Min \quad c^T x + E[Q(x, D)]$$

$$s.t. \quad x \geq 0$$

where $Q(x, D)$ is the optimal value of the second stage problem.
Second stage -

$$Min \quad (l-q)^T z - s^T y$$

$$s.t. \quad y = x - A^T z$$
$$0 \leq z \leq d$$
$$y \geq 0$$

Multistage model

Here the manufacturer has a planning horizon of T periods. The demand is modeled as a stochastic process D_t, $t = 1, ...T$ is a random vector of demands for the products. The unused parts can be stored from one period to the next, and holding one unit of part j in inventory costs h_j. $D_{[t]} = (D_1, ..., D_t)$ denotes the history of the demand process in periods $1, ..., t$. $x_{t-1} = (x_{t-1,1}, ..., x_{t-1,n})$ denotes the vector of quantities ordered at the beginning of stage t, before the demand vector D_t becomes known. The numbers of units produced in stage t will be denoted by z_t and the inventory level of parts at the end of stage t by y_t for $t = 1, ..., T$. The problem at $t = T$ is

$$Min \quad (l - q)^T z_T - s^T y_T$$
$$s.t. \quad y_T = y_{T-1} + x_{T-1} - A^T z_T$$
$$0 \leq z_T \leq d_T$$
$$y_T \geq 0$$

Let the optimal value of this problem be $Q_T(x_{T-1}, y_{T-1}, d_T)$. Then the problem at $t = T - 1, ..., 1$ can be written as

$$Min \quad (l - q)^T z_t + h^T y_t + c^T x_t + \mathcal{Q}_{t+1}(x_t, y_t, d_{[t]})$$
$$s.t. \quad y_t = y_{t-1} + x_{t-1} - A^T z_t$$
$$0 \leq z_t \leq d_t$$
$$y_t \geq 0$$

where $\mathcal{Q}_{t+1}(x_t, y_t, d_{[t]}) = E[Q_{t+1}(x_t, y_t, D_{[t+1]})|D_{[t]} = d_{[t]}]$ and optimal value $= Q_t(x_{t-1}, y_{t-1}, d_T)$ At stage t=1, the symbol y_0 represents the initial inventory levels of the parts, and the optimal value function $Q(x_0, d_1)$ depends only on the initial order x_0 and realization d_1 of the first demand D_1. The initial problem is to determine the first order quantities x_0. It can be written as

$$Min \quad c^T x_0 + E[Q_1(x_0, D_1)]$$
$$s.t. \quad x_0 \geq 0$$

2.3 Portfolio Selection

We want to invest capital W_0 in n assets, by investing an amount x_i in asset i for $i = 1, ..., n$. Each asset has a respective return rate R_i(per one period of time), which is unknown at the time we need to make our decision. We address now a question of how to distribute wealth W_0 in an optimal way. The total wealth resulting from our investment after one period of time equals

$$W_1 = \sum_{i=0}^{n} \xi_i x_i$$

where $\xi_i = 1 + R_i$

Various stochastic programming models for this problem can be formed depending on the objective function and the constraints :

Model 1 - Maximize Expected Return

One can try to maximize the expected return on an investment. This leads us to the problem:

$$\begin{aligned} &Max \quad E[W_1] \\ &s.t. \quad \sum_{i=1}^{n} x_i = W_0 \\ &\qquad x \geq 0 \end{aligned}$$

We have here

$$E[W_1] = \sum_{i=1}^{n} E[\xi_i]x_i = \sum_{i=1}^{n} \mu_i x_i$$

This problem has a simple optimal solution of investing everything into an asset with the largest expected return rate and has the optimal value of $\mu^* W_0$ where $\mu^* = max_{1 \leq i \leq n} \mu_i$

Model 2 - Maximize Expected Utility

Here we maximize the expected utility of the wealth represented by a concave non-decreasing function $U(W_1)$. This leads to the following optimization problem :

$$Max \quad E[U(W_1)]$$

$$s.t. \quad \sum_{i=1}^{n} x_i = W_0$$
$$x \geq 0$$

Let $U(W)$ be defined as:

$$U(W) = (1+q)(W-a) \; if \; W \geq a$$
$$(1+r)(W-a) \; if \; W \leq a$$

with $r > q > 0$ and $a > 0$ This problem can be formulated as the following two stage problem :

$$Max \quad E[Q(x,\xi)]$$
$$s.t. \quad \sum_{i=1}^{n} x_i = W_0$$
$$x > 0$$

where $Q(x,\xi)$ is the optimal value of

$$Max \quad (1+q)y - (1+r)z$$
$$s.t. \quad \sum_{i=1}^{n} \xi_i x_i = a + y - z$$

Model 3 - Constraining the variance

Yet another approach can be to maximize the expected return while controlling the involved risk of the investment. Here we evaluate risk by variability of W measured by its variance. Since W_i is a linear function of variables ξ_i, we have

$$Var[W_1] = x^T \Sigma x = \sum_{i,j=1}^{n} \sigma_{ij} x_i x_j$$

where $\Sigma = [\sigma_{ij}]$ is the covariance matrix of the random vector ξ. For a specified constant $\nu > 0$ the problem can be written as:

$$Max \quad \sum_{i=1}^{n} \mu_i x_i$$
$$s.t. \quad \sum_{i=1}^{n} x_i = W_0$$
$$x^T \Sigma x \leq \nu$$
$$x \geq 0$$

Model 4 - Minimize variance while constraining the expected return

Another possible formulation is to minimize $Var[W_1]$, keeping the expected return $E[W_1]$ above a specified value tau. That is,

$$\begin{aligned} Min \quad & x^T \Sigma x \\ s.t. \quad & \sum_{i=1}^{n} x_i = W_0 \\ & \sum_{i=1}^{n} \mu_i x_i \geq \tau \\ & x \geq 0 \end{aligned}$$

Model 5 - Impose Chance constraints

We can also approach risk control by imposing chance constraints:

$$Max \quad \sum_{i=1}^{n} \mu_i x_i$$

$$s.t. \sum_{i=1}^{n} x_i = W_0$$

$$Pr\{\sum_{i=1}^{n} \xi_i x_i \geq b\} \geq 1 - \alpha$$

$$x \geq 0$$

Suppose ξ has a multivariate normal distribution with mean vector μ and covariance matrix $\Sigma.W_1$ then has a normal distribution with mean $\sum_{i=1}^{n} \mu_i x_i$ and variance $x^T \Sigma x$ and

$$Pr(W_1 \geq b) = Pr(Z \geq \frac{b - \sum_{i=1}^{n} \mu_i x_i}{\sqrt{x^T \Sigma x}}) = \Phi(\frac{\sum_{i=1}^{n} \mu_i x_i - b}{\sqrt{x^T \Sigma x}})$$

where $Z\ N(0,1)$ and $\Phi(z) = Pr(Z \leq z)$ is the cdf of Z. Therefore we can write the chance constraint in the form

$$b - \sum_{i=1}^{n} \mu_i x_i + z_\alpha \sqrt{x^T \Sigma x} \leq 0$$

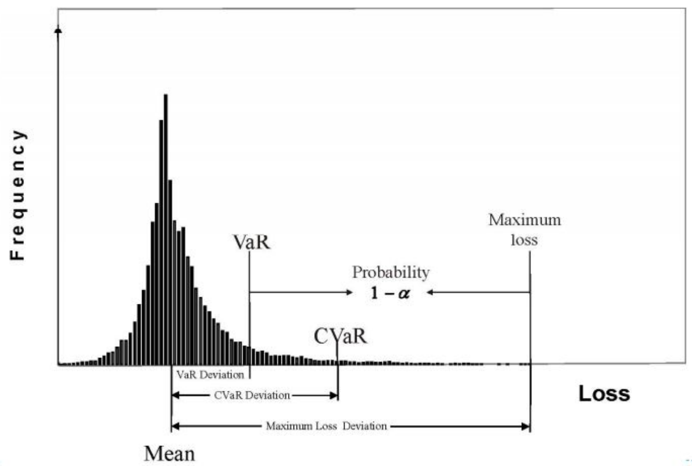
where $z_\alpha = \Phi^{-1}(1 - \alpha)$ is the $1 - \alpha$-quantile of the standard normal distribution. If $0 \leq \alpha \leq \frac{1}{2}$ then $z_\alpha \geq 0$ and the constraint is convex.

Model 6 - Minimize CVaR

Yet another approach can be to minimize the Conditional Value-at-Risk (CVaR) [3]. Let w be a given portfolio and $1-\alpha$ be the given confidence level. Then $CVaR_{1-\alpha}(w)$ associated with the portfolio w is defined as:

$$CVaR_{1-\alpha}(w) = \frac{1}{\alpha} \int_{f(w,r) \geq VaR_{1-\alpha}(w)} f(w,r)p(r)dr$$

where $f(w,r)$ is the loss function.



Chapter 3

Implementations

All the data for the above models was generated using quandl API. After fetching data for the stock prices for past 6 years from 31-01-2010 to 31-12-2016 with 1 month time period, we generated the return series, mean series and variance-covariance matrix for the past data. We have used cvx package in Matlab for solving the obtained optimization problems.

All the code files are available at: `https://github.com/raunaak/StochasticProgramming`

Table 3.1: MEAN SERIES

NSE/RELIANCE	EOD/GS	NSE/MARUTI	EOD/MSFT	NSE/TCS
1.002833615	1.009067	1.021046685	1.011722	1.01591

Table 3.2: VARIANCE SERIES

	NSE/RELIANCE	EOD/GS	NSE/MARUTI	EOD/MSFT	NSE/TCS
NSE/RELIANCE	0.004889819	0.001773	0.003748954	0.0012848	0.000244
EOD/GS	0.001772609	0.006672	0.002183366	0.0024767	0.001125
NSE/MARUTI	0.003748954	0.002183	0.009725475	0.001926	-0.00067
EOD/MSFT	0.001284768	0.002477	0.001925973	0.0043629	0.000503
NSE/TCS	0.000243549	0.001125	-0.00066542	0.000503	0.003615

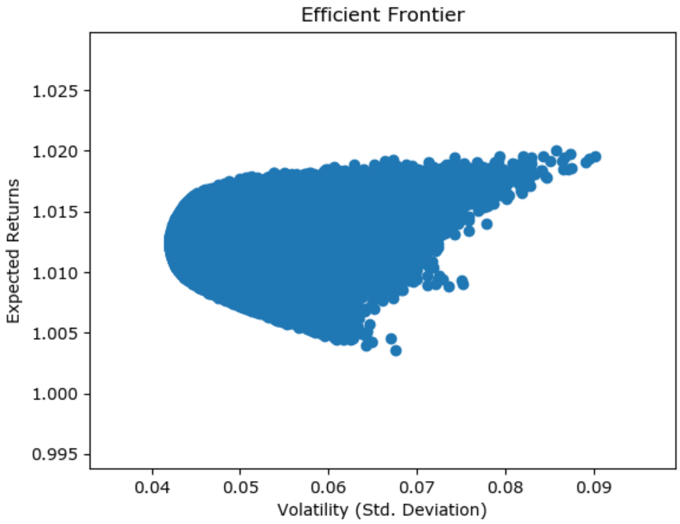


Figure 3.1: Markowitz Curve

The obtained results follow the Markowitz Curve. We generated these curves using Monte Carlo simulation. The above figure is obtained using 5,00,000 different weights.

3.1 Static Models

Model 1 - Expectation Maximization

In this model, we tried to maximize expectations subject to a variance threshold. Mathematically,

$$Max \quad \sum_{i=1}^{n} \mu_i x_i$$

$$s.t. \quad \sum_{i=1}^{n} x_i = W_0$$

$$x^T \Sigma x \leq \nu$$

$$x \geq 0$$

Each table contains a dark colored column which is further used for testing the formulated models.

Table 3.3: Expectation Maximization Results

ν	0.001	0.002	0.003	**0.004**	0.005	0.01
NSE/RELIANCE	NaN	3.50E-07	3.31E-08	**3.12E-08**	4.48E-08	9.13E-09
EOD/GS	NaN	6.03E-07	6.36E-08	**5.80E-08**	8.26E-08	1.38E-08
NSE/MARUTI	NaN	0.24794	0.499594	**0.62546**	0.71545	0.99999
EOD/MSFT	NaN	0.19120	1.20E-07	**9.52E-08**	1.30E-07	1.77E-08
NSE/TCS	NaN	0.56085	0.50040	**0.37453**	0.28454	3.43E-08
Optimal Value	NaN	1.01638	1.01848	**1.01912**	1.01959	1.02105

Model 2 - Variance Minimization

In this model, we tried to maximize expectations subject to a variance threshold. Mathematically,

$$
\begin{aligned}
&Min \quad x^T \Sigma x \\
s.t. \quad &\sum_{i=1}^{n} x_i = W_0 \\
&\sum_{i=1}^{n} \mu_i x_i \geq \tau \\
&x \geq 0
\end{aligned}
$$

Table 3.4: Variance Minimization Results

τ	**1.01**	1.015	1.02	1.021	1.022
NSE/RELIANCE	**0.20643**	6.21E-02	8.59E-10	4.10E-09	NaN
EOD/GS	**0.01576**	8.35E-06	1.61E-09	7.73E-09	NaN
NSE/MARUTI	**0.08016**	0.17788	0.79622	0.99091	NaN
EOD/MSFT	**0.24773**	0.24156	2.50E-09	1.19E-08	NaN
NSE/TCS	**0.44990**	0.51843	0.20377	0.00908	NaN
Optimal Value	**0.00176**	0.00185	0.0061	0.0095	Inf

Model 3 - Utility function

Suppose we aim for some target wealth to be achieved given initial wealth with time t. If we are able to achieve target wealth, the rest of the money is saved in bank with interest rate 6% else we need to borrow money from the bank with higher interest rate 10%. We assumed x to be multivariate normally distributed and generated different samples. We maximize our empirical cost function, i.e., the sum of individual cost functions generated for each sample. Theoretically,

$$Max \quad E[U(W_1)]$$

$$s.t. \quad \sum_{i=1}^{n} x_i = W_0$$

$$x \geq 0$$

$$U(W) = \begin{cases} \text{(1+q)(W - a)} & \text{if } W \geq a \\ \text{(1+r)(W - a)} & \text{if } W \leq a \end{cases}$$

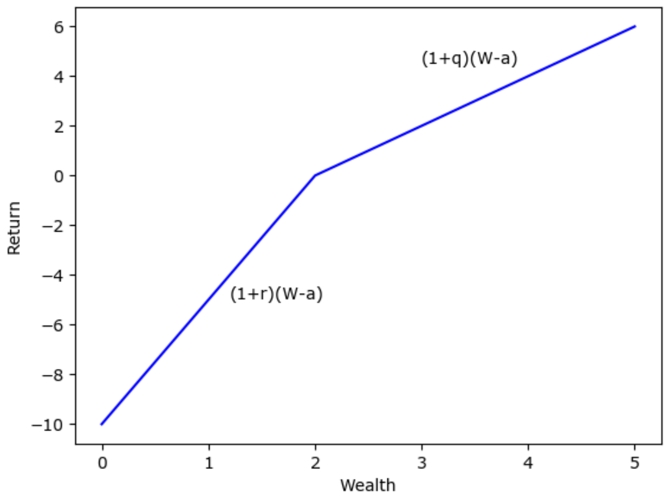


Figure 3.2: Piecewise Concave Utility Function

The results for 10 sample case is shown below with initial wealth = 1 and required wealth = 1.01. We see that the above function behaves like a vanilla expectation maximization function. But, if the slope r is too high as compared to q then we obtain values as shown in the last column (particularly, r = 1, q = 10).

Table 3.5: Utility Function Results

NSE/RELIANCE	**6.83E-10**	3.84E-09	3.84E-09	1.03E-09	3.39E-10
EOD/GS	**3.10E-09**	5.09E-09	5.09E-09	1.06E-09	0.5722542276
NSE/MARUTI	**1**	1	1	1	0.4377457712
EOD/MSFT	**4.27E-10**	3.14E-09	3.14E-09	9.97E-10	3.05E-10
NSE/TCS	**5.41E-10**	3.61E-09	3.61E-09	1.02E-09	5.51E-10
Optimal Value	**1.056**	0.2091	0.4184	1.1546	0.188878
Variance	**0.00972**	0.15	1	10	1

Model 4 - Chance Constraints

We introduced chance constraint in our model as follows:

$$Pr(\sum_{i=1}^{n} \epsilon_i x_i \geq b) \geq 1 - \alpha$$

With the multi-variate normal assumption of returns, the problem reduces down to:

$$Max \quad \sum_{i=1}^{n} \mu_i x_i$$

$$s.t. \sum_{i=1}^{n} x_i = W_0$$

$$b \sum_{i=1}^{n} \mu_i x_i + z_\alpha \sqrt{x^T \Sigma x} \leq 0$$

$$x \geq 0$$

Table 3.6: Chance Constraint Results

	0.01, 1.0	0.3, 1.0	**0.35, 1.0**	0.4, 1.0
NSE/RELIANCE	NaN	NaN	**5.14E-09**	2.87E-08
EOD/GS	NaN	NaN	**1.35E-08**	6.02E-08
NSE/MARUTI	NaN	NaN	**0.53282**	0.99582
EOD/MSFT	NaN	NaN	**3.69E-08**	1.01E-07
NSE/TCS	NaN	NaN	**0.46717**	0.0041
Optimal Value	NaN	NaN	**1.01865**	0.0095

We also verified our assumption on multi-variate normality assumption using Skew and Kurtosis using ICS library in R. Kurtosis rejected our multi-variate hypothesis with small p-value of 0.0002 while Skewness did not reject our hypothesis with p-value of 0.1363.

Model 5 - CVaR Minimization

We have implemented the CVaR minimization model using the Financial Toolbox in Matlab. The model solved is:

$$Min \quad CVaR_{1-\alpha}(x)$$

$$s.t. \sum_{i=1}^{n} x_i = 1$$

$$x \geq 0$$

Table 3.7: CVaR Results

α	0.75	0.80	0.85	0.90	**0.95**
NSE/RELIANCE	0.080952	0.093474	0.10393	0.11178	**0.14296**
EOD/GS	0.0053597	0.001782	3.6044e-21	0.012519	**1.4772e-17**
NSE/MARUTI	0.16541	0.15398	0.14603	0.1508	**0.12676**
EOD/MSFT	0.23675	0.25105	0.25157	0.23643	**0.2476**
NSE/TCS	0.51153	0.49972	0.49846	0.48847	**0.48268**
Expected Return	1.0143	1.0146	1.0144	1.0141	**1.0137**

3.2 Two stage models

We have used **Benders decomposition** to solve two stage stochastic programming problems. Consider the following two stage stochastic programming problem [4]:

$$Min \quad c^T x + Q(x)$$

$$Ax = b$$

$$x \geq 0$$

where

$$Q(x) = \sum_j p_j Q(x, \xi_j)$$

$$Q(x, \xi_j) = min\, q(\xi_j)^T y$$

$$s.t. \quad Wy = h(\xi_j) - T(\xi_j)x$$

$$y \geq 0$$

Here we have assumed that the random variable ξ has a discrete distribution where $Pr(\xi = \xi_j) = p_j$ This problem can also be formulated as:

$$min \quad c^T x + \theta$$

$$s.t. \quad Ax = b$$

$$\theta \geq Q(x)$$

$$x \geq 0$$

where $Q(x)$ is defined as before.

3.2.1 Bender's decomposition

Benders decomposition can be divided into two stages -

1. Feasibility cut
2. Optimality cut

Feasibility cut

Feasibility cut is a constraint on the first stage variables. Given a solution $x = x_0$ of the first stage problem, we check if this solution yields second stage problems with non-empty feasible region for all possible realizations of ξ. If for any realization of ξ the second stage problem becomes infeasible, we generate a feasibility cut so that it removes x_0 from the feasible region of the first stage problem. To find the feasibility cut we solve the following linear programming problem:

$$max_\sigma \sigma(h(\xi) - T(\xi)x_0) | \sigma^T W \leq 0, ||\sigma|| \leq 1$$

where $||\sigma||$ is the sum of all the components of the vector σ. If for some realization of ξ say ξ_0 we find a positive optimal value of the problem then we generate the feasibility cut as:

$$\sigma^T(h(\xi_0) - T(\xi_0)x) \leq 0 \iff \sigma^T T(\xi_0)x \geq \sigma^T h(\xi_0)$$

Now our modified first stage problem is:

$$min \quad c^T x + \theta$$

$$s.t. \quad Ax = b$$

$$\sigma^T(\xi_0)x \geq \sigma^T h(\xi_0)$$

$$x \geq 0$$

We then find a feasibility cut for this problem(if possible). We keep on finding feasibility cuts until it is not possible to find any more feasibility cut.

Optimality cut

Optimality cuts are generated only after all the feasibility cuts are found. The idea of the optimality cut is to gradually take us towards the optimal solution. Our original problem is a minimization problem. So we intend to find a lower bound of θ. Optimality cuts can therefore be looked upon as lower bounds of θ. To find the optimality cut we solve the following problem:

$$min \quad c^T x + \theta$$

$$s.t. \quad Ax = b$$

$$\gamma_k^T x \geq \delta_k$$

$$x \geq 0$$

Let's call this optimization problem (A). Let the optimal solution of this problem be $x = x_0$. We choose the initial value of θ as $-\infty(\theta_0)$. Then we check whether $\theta_0 \geq Q(x_0)$ or not. If this is true then x_0 is the optimal solution of the two stage problem else we generate optimality cut as:

$$\theta \geq \sum_j p_j(\hat{\pi}_j)[h(\xi_j) - T(\xi_j)x] = \alpha + \beta^T x$$

where $\hat{\pi_j}$ is the optimal solution of the dual problem of the second stage problem for the jth scenario.Let's call this constraint (1). The dual problem for the jth scenario is:

$$max \quad \pi^T(h(\xi_j) - T(\xi_j)x_0)|\pi^T W \leq q(\xi_j)^T$$

Now we add the constraint (1) to the optimization problem (A) and solve it. Suppose we get the optimal solutions as x_1 and θ_1. Now we check if $\theta_1 \geq Q(x_1)$ or not and repeat the above steps till we get an optimal solution. It may happen that θ and $Q(x)$ are very close to each other and keep getting closer after each iteration but $\theta \leq Q(x)$ every time we check. So in addition to $\theta \geq Q(x)$ we also check if the difference between θ and $Q(x)$ is small or not. One disadvantage of the Benders Decomposition is that we dont know beforehand after how many iterations we will get the optimal solution.

Algorithm for Benders Decomposition Method:

1. Initialize the no. of feasibility cuts, $K = 0$, the no. of optimality cuts,L=0 and $\theta = -\infty$.
2. If the problem $\{min\, L^T x | Ax = b, x \geq 0\}$ is not feasible, assign feasible = FALSE or else assign feasible =TRUE and let $x = \hat{x}$ be the optimal value of x.
3. Assign stop = !(feasible).
4. If stop = FALSE then take $K = K + 1$ and find the feasibility cut.
5. If no new feasibility cuts, find $Q(\hat{x})$ and assign stop=TRUE if $\hat{\theta} \geq Q(\hat{x})$, else stop = FALSE.
6. If stop = TRUE, stop the program and $Q(\hat{x})$ is the final optimal solution.
7. If stop=FALSE, increase $L = L + 1$ and find the optimality cut.
8. If stop = FALSE and $L > 0$ solve the following problem:

$$min \quad c^T x + \theta^+ - \theta^-$$

$$s.t. \quad Ax = b$$

$$-\Gamma\hat{x} \geq \Delta$$

$$-\beta\hat{x} + \theta^+ - \theta^- \geq \alpha$$

$$\hat{x}, \theta^+, \theta^- \geq 0$$

9. If stop = FALSE and $L = 0$ solve the following problem:

$$min \quad c^T x + \theta^+ - \theta^-$$

$$s.t. \quad Ax = b$$

$$-\Gamma \hat{x} \geq \Delta$$

$$\hat{x} \geq 0$$

10. For the problem in step (8) if feasible = TRUE then $\hat{\theta} = \theta^+ - \theta^-$ and for the problem in step (9) if feasible = TRUE then $\hat{\theta} = -\infty$.

11. Assign stop = !(feasible).

12. Repeat steps (4) - (11) while stop = FALSE.

3.2.2 A Multi-product assembly problem

A furniture company has 3 types of input namely Lumber, Finishing and Carpentry. Cost of each input is given. They have to make desks, tables and chairs and sell them to make a profit. However there can be 3 types of demand scenarios possible Low, Most Likely and High. At the time of buying input the manager of the company does not know what kind of demand will be there. So he buys some input and after observing the demand scenario that has occurred he now tries to make desks, tables and chairs so as to maximize their profit.

Table 3.8: Cost of input factors and input factor requirements

Resource	Costs	Input Requirements		
	$	Desk	Table	Chair
Lumber(bd ft)	115	105	120	116
Finishing (hrs)	80	97	85	65
Carpentry (hrs)	95	105	75	107

Table 3.9: Demand scenarios and Sell Prices

Demand Scenarios				Sell price
	Low	Most likely	High	$
Desks	50	150	250	60
Tables	20	110	250	40
Chairs	200	225	500	10
Probability	0.3	0.4	0.3	

Solution : Two-stage formulation of the problem -
First Stage-

$$min\ (2\ 4\ 5.2)^T(x_l\ x_f\ x_c) + \theta$$

$$s.t.\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ x_f \\ x_c \end{bmatrix} \leq \begin{bmatrix} 3500 \\ 1500 \\ 875 \end{bmatrix}$$

$$\theta \geq Q(x)$$

$$(x_l\ x_f\ x_c) \geq 0$$

Second Stage -

$$Q(x, \xi) = min\ (-60\ -40\ -10)^T(y_d\, y_t\, y_c)$$

$$s.t.\quad \begin{bmatrix} 8 & 6 & 1 \\ 4 & 2 & 1.5 \\ 2 & 1.5 & 0.5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_d \\ y_t \\ y_c \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ d_d(\omega_s) \\ d_t(\omega_s) \\ d_c(\omega_s) \end{bmatrix} - \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_l \\ x_f \\ x_c \end{bmatrix}$$

$$(y_d\ y_t\ y_c) \geq 0$$

x_l, x_f, x_c : amount of labour, finishing and carpentry bought.
y_d, y_t, y_c : no. of desks, tables and chairs made.
$d_d(\omega_s), d_t(\omega_s), d_c(\omega_s)$: demands of desks, tables and chairs respectively for a particular scenario.
Using the Bender's decoomposition we get the solution as : 1300 bd ft of lumber,

540 hours of finishing and 325 hours of carpentry. The profit will be approximately \$ 1730 on an average.

3.2.3 An Asset Liability Management Problem

We have 3 kinds of assets A, B and C and a capital of Rs. 15000. We want to divide this capital into the 3 assets at the first time point t_0 in such a way that when we sell off some assets at time point t_1 our sale proceeds is maximum.to prevent investing all our capital into buying one asset we put a restriction that that we wont invest more than Rs.7000 in any asset.There will be 3 types of scenarios possible at t_1 where the prices of the assets will change. We make an assumption that at t_1 we will be selling off only the profitable assets. There is also a transaction cost of 5%. The data is given below:
(S_1,S_2 and S_3 are the three possible scenarios)

Table 3.10: The prices given here are price in Rs. per 100 face value amount of each asset

Asset	Price(t_0)	Price(t_1)		
		S_1	S_2	S_3
A	115	105	120	116
B	80	97	85	65
C	95	105	75	107
Prob.		0.3	0.4	0.3

Solution : Two-stage formulation of the problem -
First stage :

$$min(0\ 0\ 0\ 0\ 0\ 0)^T(x_A\ x_B\ x_C\ z_1\ z_2\ z_3) + \theta$$

$$s.t. \quad 1.05 * \begin{bmatrix} 115 & 80 & 95 & 0 & 0 & 0 \\ 115 & 0 & 0 & 1 & 0 & 0 \\ 0 & 80 & 0 & 0 & 1 & 0 \\ 0 & 0 & 95 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \\ x_C \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 15000 \\ 7000 \\ 7000 \\ 7000 \end{bmatrix}$$

$$\theta \geq Q(x)$$

$$x_A, x_B, x_C, z_1, z_2, z_3 \geq 0$$

where $Q(x) = 0.3Q(x, S_1) + 0.4Q(x, S_2) + 0.3Q(x, S_3)$

Second stage :
For a particular scenario say S_1 :

$$Q(x, S_1) = min\ 0.95 * (105\ -97\ -105\ 0\ 0\ 0)^T(y_A\ y_B\ y_C)$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_A \\ y_B \\ y_C \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x_A} \\ \bar{x_B} \\ \bar{x_C} \\ \bar{z_1} \\ \bar{z_2} \\ \bar{z_3} \end{bmatrix}$$

x_A, x_B, x_C : amount of assets A,B and C to buy at time point t_0 y_A, y_B, y_C : amount of assets A,B and C that we have to sell at time point t_1 $z_1, z_2, z_3, s_1, s_2, s_3$: slack variables $\bar{x_A}, \bar{x_B}, \bar{x_C}, \bar{z_1}, \bar{z_2}, \bar{z_3}$: a particular solution of the 1st stage problem
Using Bender's decomposition we get the solution : Buy asset A worth Rs.7003.5, asset B worth Rs.6972 and asset C worth Rs.997.5. The maximum average sale proceeds come out to be Rs.10161.

3.3 Multi-Stage Models

3.3.1 Multi-Stage Minimum Return Soft Optimization

One of the possible formulations of a model could be to minimize value at risk subject to minimum return. Our optimization problem becomes:

$$\min_{x \in X} \lambda(x, \alpha) : r(x) \geq \mu$$

where $\lambda(x, \alpha)$ represents value at risk.

Lets add some terminology to make the problem concrete [5]. Let n be the number of securities acceptable for investment, and let $x_i(j)$ be the investment ratio of security i at stage j (i = 1, 2, ..., n; j = 0, 1, ..., T - 1). Let x(j) be a vector consisting of the investment ratios at stage j, that is,

$$x(j) = (x_1(j), ..., x_n(j))$$

, then x(j) belongs to the following set:

$$X(j) = \{(x_1(j), , x_n(j))^{`} \in R^n | \sum_{i=1}^{n} x_i(j) = 1, x_i(j) \geq 0, i = 1, ..., n\}$$

Denote the matrix composed by the investment ratio vectors at all stages by x, that is,

$$x = (x(0), x(1), , x(T-1)) \in R^{n*T}$$

Let the price of security i at time t_j be $p_i(t_j)$, and let the value of the portfolio at time t_j be $W(t_j)$. Since,

$$W(t_j) = \sum_{i=1}^{n} \frac{x_i(j-1)W(t_j-1)}{p_i(t_j-1)} p_i(t_j) = W(t_{j-1}) \sum_{i=1}^{n} x_i(j-1) \frac{p_i(t_j)}{p_i(t_{j-1})}$$

$$R(x) = \frac{W(t_T)}{W(t_0)} - 1 = \frac{W(t_T)}{W(t_{T-1})} \frac{W(t_{T-1})}{W(t_{T-2})} ... \frac{W(t_1)}{W(t_0)} - 1$$

$$R(x) = \prod_{j=1}^{T} (\sum_{i=1}^{n} x_i(j-1) \frac{p_i(t_j)}{p_i(t_{j-1})}) - 1$$

$$R(x) = g(x, P) - 1$$

Return measure of strategy x is defined as $r(x) = E_P[R(x)] = E_P[g(x, P)] - 1$. The loss rate L(x) is defined as $L(x) = -R(x)$

The Value-at-Risk with respect to L(x) is:

$$\lambda(x, \alpha) = inf\{\lambda | Pr\{L(x) \leq \lambda\} \geq \alpha\}$$

Our optimization problem was $\min_{x \in X} \lambda(x, \alpha) : r(x) \geq \mu$. But, VaR is non-convex and non-smooth. So, conventional methods won't work.

Soft Optimization

The soft approach solves an optimization model by softening the goal of solving a model as follows: 1. Instead of searching for an optimal solution, the soft approach seeks a good enough solution. The top k% solution are taken as good enough solution,

where k is set according to problem requirement. 2. Instead of requiring the solution to definitely be an optimal solution, the soft approach accepts a solution if it is highly likely that the solution is a good enough solution.
Let an optimization model be
$minf(x) : x \in X^f, x \in R^n, X^f$ is feasible set
The soft approach uses the following two-stage process to generate a solution.
Stage 1: Sample the feasible set X^f to generate a finite subset S. For the sample set S, the probability with which S contains at least one good enough solution is required to be high. Let $G \subset X^f$ be a set of good enough solutions, say the top 1% solutions. Then, the following condition is required to be satisfied:

$Pr\{|S \cap G| \geq 1\} \geq 99\%$

Consequently, the best alternative in S will be a good enough solution with a probability larger than 99%.

Stage 2: Select the best alternative $x\star$ from the sample set S. Here, $x\star$ is the solution that the soft approach is looking for.

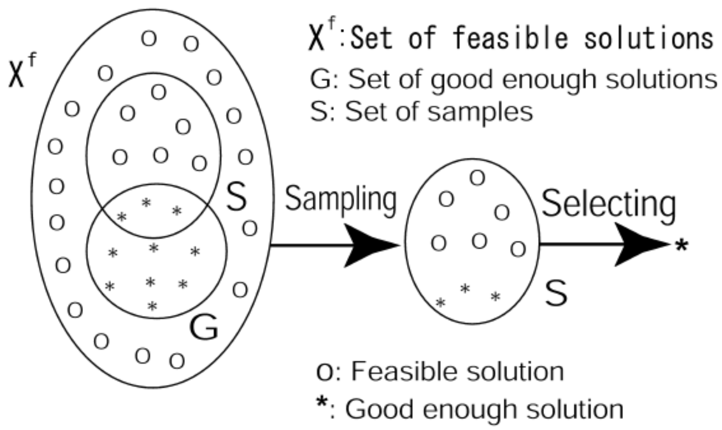


Figure 3.3: Two stage process of Soft Approach

The soft approach technique for our problem can be divided into 2 stages:

Stage 1: Create a set of 1000 sample strategies which satisfies the problem constraints.

$$X^1 = \{x \in X | E_P[g(x, P)] - 1 \geq \mu\}$$

To produce uniform samples for set X, we first generate n numbers between 0 and 1, and divide them by their unit norm. Repeating this process D (time period) times gives a sample strategy for X.

Let t_j^k be the corresponding time of t_j in the k^{th} historical group, and let $q_i(t_j^k)$ be the price of security i at time t_j^k. A matrix is formed using these price data in each historical group as follows:

$$Q^k = \begin{bmatrix} q_1(t_1^k) & q_1(t_2^k) & \dots & q_1(t_{T-1}^k) & q_1(t_T^k) \\ q_2(t_1^k) & q_2(t_2^k) & \dots & q_2(t_{T-1}^k) & q_2(t_T^k) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_n(t_1^k) & q_n(t_2^k) & \dots & q_n(t_{T-1}^k) & q_n(t_T^k) \end{bmatrix}$$

For m observations and D time period, $E_P[g(x, P)]$ could be estimated as:

$$E_P[g(x, P)] = \sum_{k=1}^{m-D+1} g(x, Q_k)$$

Stage 2: Determine the sample strategy with the least VaR

We use historical simulation method. For each $x \in X_1$, generate time-series data of its loss rate using historical prices of the securities.

$$L^k(x) = 1 - g(x, Q^k), k = 1, 2, ..., m - D + 1$$

Sort the losses obtained for a given strategy x and choose the strategy with smallest $\lambda(x, \alpha) = L^{(m-D)*\alpha}(x)$.

With the hyper parameters, = 0.95 and = 0.05, the following weights are obtained with expected return in a month = 0.0506

Table 3.11: Multi Stage Minimum Return Results

Time Period	0	1	2	3
NSE/RELIANCE	0.169405	0.271794	0.316299	0.022173
EOD/GS	0.004387	0.036054	0.136329	0.144516
NSE/MARUTI	0.163146	0.312296	0.040541	0.284698
EOD/MSFT	0.369359	0.213963	0.25217	0.221357
NSE/TCS	0.293702	0.165892	0.25466	0.327255

3.3.2 Multi Stage Controlled VaR Soft Optimization

The aim is to maximize return subject to controlled value at risk.

$$\max_{x \in X} r(x) : \lambda(x, \alpha) \leq \beta$$

Since, dealing with VaR is difficult, we would be using soft optimization techiques. Denote feasible set as:

$$X^2 = \{x \in X | \lambda(x, \alpha) \leq \beta\}$$

Stage 1: Generate 1,000 sample strategies uniformly We estimate $\lambda(x, \alpha)$ using the historical simulation method until our feasible set has 1000 samples.

Stage 2: Determine the sample strategy with the largest return The returns could be estimated using $E_P[g(x, P)] = \sum_{k=1}^{m-D+1} g(x, Q_k)$ as mentioned above. After obtaining the returns of all of the samples, the sample strategy with the largest return is selected as the solution.

The expected return for $\alpha = 0.95$ and $\beta = 0.1$ in a month is 0.0615 with weights:

Table 3.12: Multi Stage Controlled VaR Results

Time Period	0	1	2	3
NSE/RELIANCE	0.042409505	0.125266863	0.000533702	0.104495231
EOD/GS	0.001447582	0.122846935	0.146846067	0.037387058
NSE/MARUTI	0.225348595	0.24337832	0.509333427	0.29973329
EOD/MSFT	0.275083533	0.289575621	0.213048959	0.264337767
NSE/wTCS	0.455710786	0.218932261	0.130237844	0.294046654

3.3.3 Multi-Period Portfolio Optimization Using Scenario Tree

Suppose initial wealth is 60 and target wealth is 80. There are two assets, say, stocks and bonds; hence, (number of assets) I = 2. Assume up branches return is 1.25 for stocks and 1.14 for bonds and (bad) down branches, total return is 1.06 for stocks and 1.12 for bonds. Bonds play the role of safer assets here. Returns are assumed to be a sequence of i.i.d. random variables. A piecewise concave utility function is chosen such that the reward rate q for excess wealth above the target liability is 1 and the penalty rate r for the shortfall below the target liability is 4. We assume each branch at each node is equiprobable, i.e., the conditional probabilities are always 0.5. The final nodes have probability $p_s = 1/8$, for s = 1, ...,8 [6].

The following notation and decision variables are used in the model formulation:
1. N is the set of event nodes; in our case N = n0, n1, n2, ..., n14.
2. Each node $n \in N$, apart from the root node n0, has a unique direct predecessor node, denoted by a(n): for instance, a(n3) = n0.
3. There is a set $S \cap N$ of leaf (terminal) nodes; in our case S = n7, ..., n14.
4. For each node $s \in S$, we have surplus and shortfall variables w_+^s and w_-^s, related to the difference between terminal wealth and liability.
5. There is a set $T \cap N$ of intermediate nodes, where portfolio rebalancing may occur after the initial allocation in node n0; in our case T = n1, n2, ..., n6.
6. For each node $n \in n0 \cup T$ there is a decision variable x_{in}, expressing the money invested in asset i at node n.
7. R_{in} is the total return for asset i during the period that leads to node n.

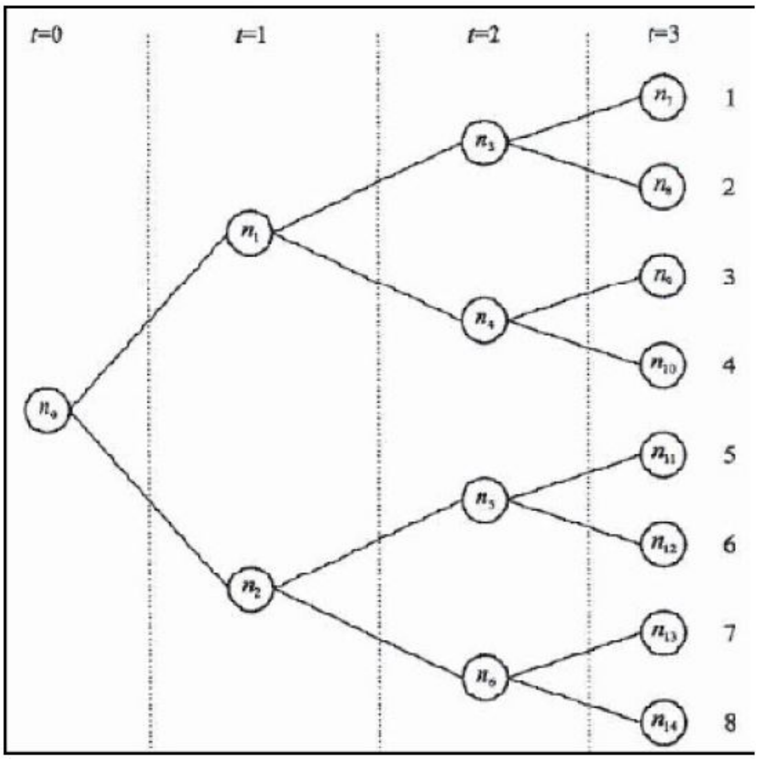


Figure 3.4: Scenario Tree for asset liability management problem

Mathematically, the problem reduces to:

$$max \sum_{s \in S} \pi^s (g w_+^s - r w_-^s)$$

$$s.t. \sum_{i=1}^{I} x_{i,n_0} = W_0$$

$$\sum_{i=1}^{I} R_{i,n} x_{i,a(n)} = \sum_{i=1}^{I} x_{in} \forall n \in T$$

$$\sum_{i=1}^{I} R_{i,s} x_{i,a(s)} = L + w_+^s - w_-^s \forall s \in S$$

$$x_{in}, w_+^s, w_-^s \geq 0$$

The obtained results are shown below:
When there is less difference between reward and penalty, all the money is invested in the asset with higher expected return. In our example, it is the stocks. When there is huge penalty as compared to reward for achieving the target, the investor tries to diversify its portfolio.

Table 3.13: Multi Stage Scenario Tree Results

	r, g = 5, 2		r, g = 15, 2	
Nodes	Stocks	Bonds	Stocks	Bonds
n0	60.00000003	-2.71E-08	42.63006751	17.36993249
n1	74.99999994	6.63E-08	73.08930743	3.34E-10
n2	26.97907396	36.62092605	16.17813391	48.46406204
n3	93.74999996	3.57E-08	91.36163428	7.66E-10
n4	79.49999991	3.57E-08	77.47466587	7.07E-10
n5	75.4716981	3.85E-08	75.47169811	5.00E-09
n6	5.826260607	63.78699496	2.48E-10	71.42857143
Optimal value	2.2147		2.1445	

Chapter 4

Results

We trained our model from stock prices for past 6 years from 31-01-2010 to 31-12-2016 with 1 month time period. But, we need to test how our models performed. We ed our models on monthly stock price data from 31-12-2016 to 30-04-18 for 16 months. This period includes both periods of high return in the financial year '17 and bearish trend in the financial year '18. Hence, it seems reasonable to compare based on this data. For every model, we have used weights which are highlighted in its table.

Table 4.1: Stock Prices

Month	NSE/RELIANCE	EOD/GS	NSE/MARUTI	EOD/MSFT	NSE/TCS
0	1	1	1	1	1
3	1.334864141	0.975928833	1.106985622	1.058932715	1.019441205
6	1.545350172	0.982600733	1.314849217	1.124516628	1.117499327
9	0.900162648	1.057387057	1.393094965	1.286620263	1.176787156
12	0.919728282	1.168192918	1.613385927	1.469605568	1.395797829
15	0.953214696	1.04696494	1.49015566	1.45800464	1.548098484

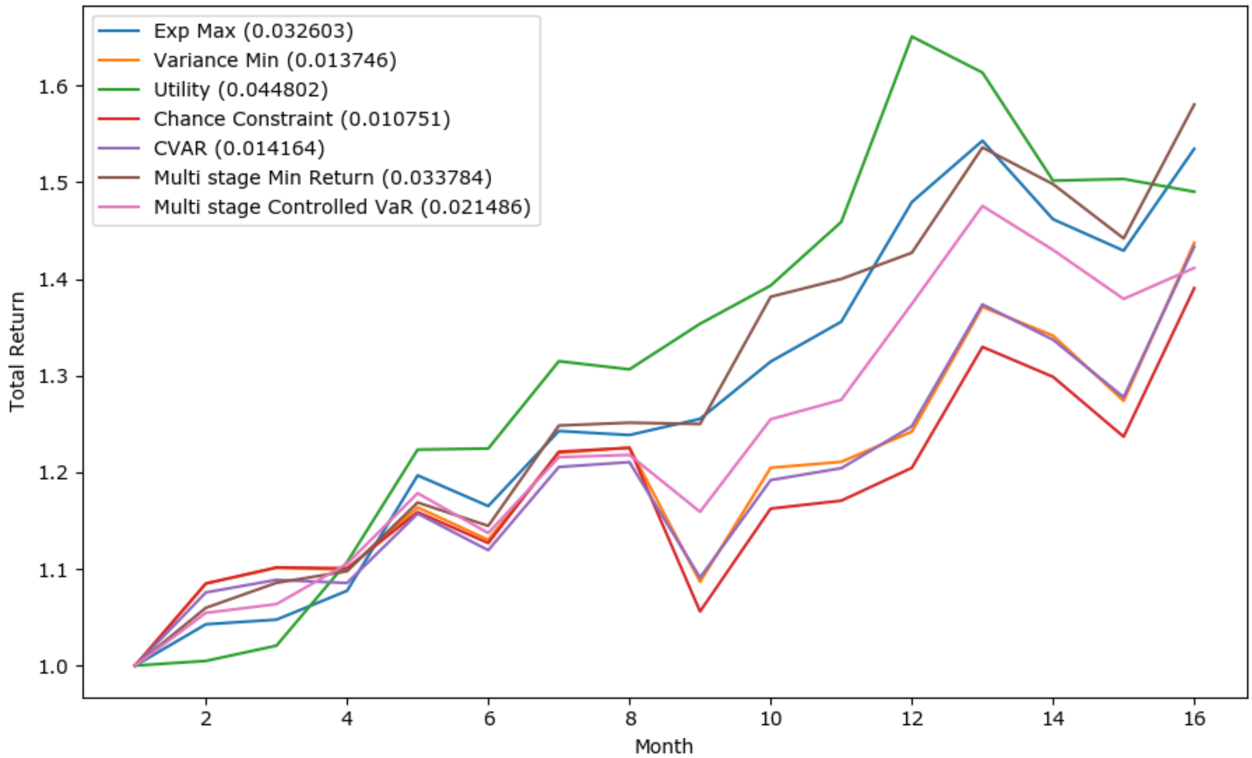


Figure 4.1: Performance Analysis

Following are some observations from the figure: 1. Utility Function Model (Static Model 3) is the most volatile. It gave excellent result when the market was performing well and reduced heavily in the bearish season. Hence, in the long run, it failed to give maximum returns.
2. Expectation Maximization Model (Static Model 1) is composed of 2 stocks and is less volatile than the above model. In the long run, it is able to give better results than highly volatile portfolio.
3. Variance Minimization Model (Static Model 2) has very low variance compared to the above models. During the bearish phase of the market, it had comparatively less fall. Hence, it is more suitable for risk averse investor.
4. Chance Constraint Model (Static Model 4) has the lowest variance among all the models and thus, the least returns. It is constantly following the same path as Variance Minimization model with some difference.
5. CVAR (Static Model 5) has variance similar to Variance Minimization model and both follow the same return series. Both are suitable for risk averse investor.

6. Multi stage Controlled VaR Model has variance in between the high volatility and low volatility models due to the choice of parameters. For most of the time, these models give returns in between high volatility and low volatility models.
7. Multi stage Minimum Return Model has high variance and seems to be optimal choice of model for risky investors. Since, there is choice to change weights between months, this is the only model which gives higher return than any individual stock.

Bibliography

[1] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[2] Alexander Shapiro. *Stochastic Programming: Modeling and Theory. Georgia Institute of Technology, Atlanta, USA*. 2013.

[3] Aparna Mehra R. Khemchandani Suresh Chandra, S. Dharmaraja. *Financial Mathematics: An Introduction*. Narosa Publishing House, 2012.

[4] B.Nandani. Bender's decomposition. 2011.

[5] Chunhui Xu, Jie Wang, and Naoki Shiba. Multistage portfolio optimization with var as risk measure. *International Journal of Innovative Computing, Information and Control*, 3(3):709–724, 2007.

[6] Mehmet Can and Narela Bajram. A stochastic programming approach for multiperiod portfolio optimization with transaction costs. *Southeast Europe Journal of Soft Computing*, 1(2), 2012.