VALUE-AT-RISK ESTIMATION FOR DYNAMIC HEDGING

YUJI YAMADA*
Graduate School of Business Sciences, University of Tsukuba
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112-0012, Japan

JAMES A. PRIMBS†
Management Science and Engineering, Stanford University
Terman Engineering Center 700-327, CA 94305-4026, USA

Received 23 March 2001
Accepted 7 May 2001

In this work, we develop an efficient methodology for analyzing risk in the wealth balance (hedging error) distribution arising from a mean square optimal dynamic hedge on a European call option, where the underlying stock price process is modeled on a multinomial lattice. By exploiting structure in mean square optimal hedging problems, we show that moments of the resulting wealth balance may be computed directly and efficiently on the stock lattice through the backward iteration of a matrix. Based on this moment information, convex optimization techniques are then used to estimate the Value-at-Risk of the hedge. This methodology is applied to a numerical example where the Value-at-Risk is estimated for a hedged European call option on a stock modeled on a trinomial lattice.

Keywords: Dynamic hedging, Wealth balance distribution, Moments, Backward equation, Convex probability bounds

1 Introduction

Option pricing and hedging theory have been the core of modern mathematical finance since the derivation of the famous Black-Scholes formula [2], which provided a theoretical value and hedging strategy for European call/put options. The key to their formula is that there exists a trading strategy which constructs a portfolio that perfectly replicates the payoff of a call (or put) option under the following two

*E-mail: yuji@gssm.otsuka.tsukuba.ac.jp
†E-mail: japrimbs@stanford.edu
assumptions: the underlying stock price follows a geometric Brownian motion, and trading may take place in continuous time. With these assumptions, it was shown that the initial value of the replicating portfolio provides the initial price of the option. Moreover, the Black-Scholes analysis demonstrated that an option can be created synthetically by dynamically trading in the underlying asset.

An important issue in dynamic hedging is the frequency of trading. Ideally, one must adjust the portfolio continuously to replicate the payoff of an option perfectly. However, continuous trading is never possible, and there always exists a hedging error, i.e., perfect replication is not possible if the market is incomplete (see e.g., [12, 22, 23]). The failure of portfolio insurance techniques based upon dynamic hedging during the crash of 1987 provides a clear example. As noted in Chapter 13 of [13], synthetically creating options on an index does not work well if the volatility of the index changes rapidly or the index exhibits large jumps. On Monday, October 19, 1987, portfolio managers who had synthetically created options found that they were unable to sell either stocks or index futures fast enough to protect their positions, and as a result, some of them sustained heavy losses. The objective of this work is to provide a new tool to analyze the risk in a dynamic hedge when the market is incomplete.

In this work, we characterize the risk in a dynamic hedge through the moments of its error distribution, and propose an efficient method for computing these moments. The computational procedure rides on top of standard lattice based pricing and hedging techniques in an incomplete market [7, 12, 22, 23]. In particular, we investigate the case of mean square optimal hedging for a European call option, although other types of options may be treated in a similar manner (see Section 3). By exploiting structure in this problem, we show that moments may be computed on the asset lattice through the backward iteration of a matrix. Using convex probability bounds, we demonstrate that the moments can be used to estimate a specified quantile of the distribution, commonly referred to as its Value-at-Risk (VaR) in financial literature [16, 19].

This paper is organized as follows: The problem formulation is given in Section 2. Section 3 provides the main result. The solution to the mean square optimal hedging problem is briefly reviewed, and then we present a recursive algorithm to compute any moment of the resulting wealth balance. Based on these moments, in Section 4 we demonstrate that bounds on the VaR of the wealth balance may be computed using convex programming. Numerical experiments illustrate the proposed methodology in Section 5 and Section 6 offers some concluding remarks.

2 Problem Formulation

We consider a market over the time interval $t \in [0, t_N]$ consisting of two basic securities, a riskless bond and a risky asset (or stock). Traders are allowed to purchase and sell these securities at the equally spaced discrete times $t_n$, $n = 0, 1, \ldots, N$ with $t_0 = 0$. We will denote the prices of the stock and the bond at the discrete times $t = t_n$ ($n = 0, 1, \ldots, N$) by $S_n$ and $B_n$, respectively. In this paper, we will model the dynamics of these underlying securities as follows:
Bond: The price of the bond will satisfy the relation
\[ B_n = RB_{n-1}, \quad n = 1, \ldots, N, \] (2.1)
with \( R := 1 + r \) where \( r > 0 \) represents a fixed interest rate.

Stock: We assume that the stock price evolves randomly on an \( L \) state lattice model (i.e., given the price of the stock at \( t = t_{n-1} \), \( n = 1, \ldots, N \), there are \( L \) possible future prices that it can take at time \( t = t_n \)). Suppose that \( u \) and \( d \) satisfy \( u > d > 0 \), then a multi-nomial lattice can be constructed by taking the \( L \) possible future states for \( S_n \) as
\[ S_n = u^{L-1} d^{l-1} S_{n-1}, \quad l = 1, \ldots, L \] (2.2)
with probabilities \( p_l, \ l = 1, \ldots, L \) satisfying \( p_1 + \cdots + p_L = 1 \). In this case, the stock may achieve \( n(L - 1) + 1 \) possible prices at time \( t = t_n, \ n = 0, \ldots, N \) given by
\[ S_n^{(k)} = u^{n(L-1)+1-k} d^{k-1} S_0, \quad k = 1, \ldots, n(L - 1) + 1. \] (2.3)

We will consider the problem of hedging a European call option through a self-financing portfolio consisting of the above securities. Hence, before proceeding, let us define the European call option to be hedged, and the replicating portfolio.

**European Call Option:** Let \( C_n \) denote the value of a European call option which is a security with payoff
\[ C_N = (S_N - K)^+ := \max (S_N - K, \ 0) \] (2.4)
at the expiration date \( t_N \), where \( K \) is the strike price. This payoff represents the option, but not the obligation, to purchase the stock at the expiration date \( t_N \) at the strike price \( K \).

**Self-financing Replicating Portfolio:** We define a portfolio to be a couple \( (\delta_n, \theta_n) \in \mathbb{R}^2 \) whose value is given by
\[ \Omega_n := \delta_n S_n + \theta_n B_n \] (2.5)
at time \( t = t_n \) (\( n = 0 \ldots N \)), where \( \delta_n \) represents the number of shares of stock and \( \theta_n \) the number of bonds held by the trader during the time interval \( t \in [t_n, \ t_{n+1}) \). Finally, we assume that the portfolio is self-financing, i.e.,
\[ \delta_{n-1} S_n + \theta_{n-1} B_n = \delta_n S_n + \theta_n B_n, \quad \forall n = 1 \ldots N. \] (2.6)

From (2.5) and (2.6), we may write the overall portfolio dynamics as
\[ \Omega_{n+1} = R\Omega_n + \delta_n (S_{n+1} - RS_n). \] (2.7)
2.1 The Wealth Balance of a Dynamic Hedge

The wealth balance of a hedged call option writer corresponds to the value of a portfolio consisting of writing and then hedging a European call option. At time \( t = 0 \), the writer receives the price of the option \( C_0 \) and constructs a self-financing replicating portfolio with initial wealth \( \Omega_0 = C_0 \). Between \( t = 0 \) and \( t_N \), the writer trades the underlying asset \( S_n \) and buys or sells \( \delta_n - \delta_{n-1} \) shares of stock, adjusting the portfolio at every time step \( t = t_n \) \((n = 1, \ldots, N - 1)\). At time \( t = t_N \), the self-financing replicating portfolio’s value is \( \Omega_N \), where

\[
\Omega_N = R^N\Omega_0 + \sum_{n=0}^{N-1} R^{N-n-1}\delta_n (S_{n+1} - RS_n),
\]

and the writer loses (or gains) the difference between \( \Omega_N \) and \( C_N \). Let \( W_N \) denote the final wealth balance of the writer of the call option at time \( t = t_N \). Then \( W_N \) may be computed as

\[
W_N = \Omega_N - C_N = R^N\Omega_0 + \sum_{n=0}^{N-1} R^{N-n-1}\delta_n (S_{n+1} - RS_n) - (S_N - K)^+. \tag{2.9}
\]

If there exists a trading strategy \( \delta_n, \ n = 0 \ldots N \) and an initial value of the portfolio \( \Omega_0 \) such that the final value of the portfolio perfectly replicates the final payoff of the call option, \( (i.e., W_N = 0) \), then the writer of the call option does not lose (or gain) any money and the hedging error is always zero. However, in general a hedging error always occurs (except in certain idealized markets such as complete markets), and the writer of the call option is exposed to the risk represented by the wealth balance distribution. As a consequence, the ability to estimate the risk in dynamic hedges can be a valuable tool.

In this paper, we measure the risk in dynamic hedges in terms of the Value-at-Risk (VaR) of the wealth balance distribution, where the VaR is a maximum allowable loss at a specified confidence level. We propose an efficient methodology to estimate the VaR which involves the following two steps:

1. Under a mean square optimal dynamic hedging strategy, we show that the moments of the wealth balance distribution may be efficiently calculated using a backward recursion on the underlying stock lattice (see Section 3).

2. We then apply a convex optimization approach to compute upper and lower bounds on the VaR given the first \( m \) moments of the wealth balance distribution (see Section 4).

Since both the moment estimation procedure and upper and lower bound problems are efficient, we conclude that the resulting methodology provides a fast and effective algorithm for estimating the risk in dynamic hedges.
3 Moment Calculations in Optimal Dynamic Hedging

In this section, we provide a method for calculating the moments of the wealth balance $W_N$ in minimum mean square error dynamic hedging. We begin by introducing an optimal hedging strategy to minimize the mean square error of the wealth balance $W_N$ when the underlying stock is modeled on a multinomial lattice. We then develop a recursion algorithm on the underlying stock lattice which computes moments of any order of the wealth balance distribution.

3.1 Optimal Hedging Strategy

We consider a hedging scheme which minimizes the mean square value of the wealth balance in the following sense:

**Mean Square Optimal Hedging (MSOH)**

\[
\begin{align*}
\text{Given} & \quad S_0, K, t_N, N \\
\text{Minimize} & \quad \mathbb{E} \left( W_N^2 \mid S_0, \Omega_0 \right) \\
\text{Subject to} & \quad \delta_n \in \mathbb{R}, \quad n = 0, \ldots, N - 1, \quad \Omega_0 \in \mathbb{R}
\end{align*}
\]

where $W_N$ is defined in (2.9). This problem has been considered extensively in the literature (see, for instance [5, 7, 8, 9, 10, 12, 22, 23], and references therein). Hence, we will not provide a full solution of it here, but merely point out properties of the solution which are relevant for our purposes.

In particular, one can show that the local hedging parameter $\delta_n \in \mathbb{R}, \ n = 0, \ldots, N - 1$ is affine in the portfolio value. That is,

\[
\delta_n = \alpha_n(S_n)\Omega_n + \beta_n(S_n), \quad n = 0, \ldots, N - 1. \tag{3.2}
\]

Without providing full justification, this result is intuitively reasonable since the portfolio dynamics are linear in $\Omega_n$, and the cost is quadratic in $\Omega_n$, indicating that the problem has a linear-quadratic structure in $\Omega_n$. In general, the solutions to linear-quadratic problems have the above structure. We will use equation (3.2) when we compute moments in Subsection 3.2.

Finally, once the optimal hedging strategy has been found, the optimal mean square error may be computed as a function of the initial portfolio value $\Omega_0$. One may then minimize the error over the choice of $\Omega_0$ to find the optimal initial portfolio value. This optimization turns out to be a simple quadratic minimization. In this paper, we will associate this optimal initial portfolio value with the “price” of the option, and assign $C_0 = \Omega_0$. Under this price, the average wealth balance satisfies $\mathbb{E} \left( W_N \mid S_0, \Omega_0 \right) = 0$ [8], and the objective function in MSOH becomes the variance of the wealth balance. Therefore, in this situation, the MSOH problem can be thought of as minimizing the risk in the hedge as measured by the variance subject to a zero mean constraint.

**Remark 3.1** It has been shown that the above “price” can lead to arbitrage opportunities (see the example by Schweizer [23]). Nevertheless, in keeping with mean-variance theory, we will continue to refer to it as a price with the possibility of abuse.
3.2 Backward Recursive Calculation of Moments

In this section we propose a recursive algorithm to find moments of the wealth balance by solving the problem backwards on a multinomial stock lattice. We show that the algorithm computes moments of any order efficiently with polynomial time computational complexity. To explain the idea, we first consider the second moment case, and then generalize to the $m$-th moment case.

3.2.1 Computation of the Second Moment

We will first write $W_N^2$ as a quadratic form in $\Omega_N$, and then use the property of nested expectations to derive a recursion for the quadratic form.\(^1\)

Let’s begin by writing $W_N^2$ as a quadratic form in $\Omega_N$.

$$W_N^2 = (\Omega_N - C_N)^2 = \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right]^T H_N \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right], \quad H_N = \left[ \begin{array}{cc} 1 & -C_N \\ -C_N & C_N^2 \end{array} \right],$$

where $C_N = (S_N - K)^+$. The second moment of the wealth balance $W_N$ is given as

$$\mathbb{E} (W_N^2 | S_0, \Omega_0) = \mathbb{E} \left( \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right]^T H_N \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right] | S_0, \Omega_0 \right).$$

Now we are ready to apply the property of nested expectations. Conditioning on $\Omega_{N-1}$ and $S_{N-1}$ gives

$$\mathbb{E} \left( \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right]^T H_N \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right] | S_0, \Omega_0 \right)$$

$$= \mathbb{E} \left( \mathbb{E} \left( \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right]^T H_N \left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right] | S_{N-1}, \Omega_{N-1} \right) | S_0, \Omega_0 \right).$$

Focusing on the inner expectation and writing $\Omega_N$ in terms of $\Omega_{N-1}$ using (2.7) and (3.2) as follows,

$$\left[ \begin{array}{c} \Omega_N \\ 1 \end{array} \right] = J_N \left[ \begin{array}{c} \Omega_{N-1} \\ 1 \end{array} \right],$$

where

$$J_N := \begin{bmatrix} R + (\alpha_{N-1}(S_{N-1})) (S_N - RS_{N-1}) & (\beta_{N-1}(S_{N-1})) (S_N - RS_{N-1}) \\ 0 & 1 \end{bmatrix}$$

\(^1\)Those familiar with the theory of Markov processes will note that we are solving the so-called backward equation [11].
leads to
\[ \mathbb{E} \left( \begin{bmatrix} \Omega_N \\ 1 \end{bmatrix}^T H_N \begin{bmatrix} \Omega_N \\ 1 \end{bmatrix} | S_{N-1}, \Omega_{N-1} \right) = \mathbb{E} \left( \begin{bmatrix} \Omega_{N-1} \\ 1 \end{bmatrix}^T J_N^T H_N J_N \begin{bmatrix} \Omega_{N-1} \\ 1 \end{bmatrix} | S_{N-1}, \Omega_{N-1} \right) = \begin{bmatrix} \Omega_{N-1} \\ 1 \end{bmatrix}^T \mathbb{E} (J_N^T H_N J_N | S_{N-1}) \begin{bmatrix} \Omega_{N-1} \\ 1 \end{bmatrix}^T. \] (3.5)

Letting
\[ H_{N-1} := \mathbb{E} (J_N^T H_N J_N | S_{N-1}) \] (3.6)
and continuing in this manner by conditioning at time \( N - 2, N - 3, \ldots \), etc., and defining the general recursion relation,
\[ H_{n-1} := \mathbb{E} (J_n^T H_n J_n | S_{n-1}), \] (3.7)
where
\[ J_n := \begin{bmatrix} R + (\alpha_{n-1} (S_{n-1})) (S_n - R S_{n-1}) (\beta_{n-1} (S_{n-1})) (S_n - R S_{n-1}) \\ 0 \end{bmatrix} \]
allows one to iterate backwards to find \( H_0 \). Once \( H_0 \) is attained, the second moment at time \( t = 0 \) is computed as
\[ \mathbb{E} \left( W_N^2 | S_0, \Omega_0 \right) = \begin{bmatrix} \Omega_0 \\ 1 \end{bmatrix}^T H_0 \begin{bmatrix} \Omega_0 \\ 1 \end{bmatrix}. \] (3.8)
which is the desired quantity.

3.2.2 The General Case

The above algorithm can readily be generalized to the higher order moment case where we are computing \( \mathbb{E} (W_N^m | S_0, \Omega_0) \). The only difference is that we use a polynomial form
\[ \begin{bmatrix} \Omega_n^h \\ \Omega_n^{h-1} \\ \vdots \\ \Omega_n \\ 1 \end{bmatrix}^T H_n \begin{bmatrix} \Omega_n^h \\ \Omega_n^{h-1} \\ \vdots \\ \Omega_n \\ 1 \end{bmatrix}, \quad n = 0, \ldots, N, \] (3.9)
to compute the \( m \)-th order moment, where \( H_n \in \mathbb{R}^{(l+1) \times (l+1)} \) and \( h := \lceil m/2 \rceil \). For example, \( H_N \) can readily be computed as a matrix satisfying
\[ (W_N)^m = (\Omega_N - C_N)^m = \begin{bmatrix} \Omega_N^h \\ \Omega_N^{h-1} \\ \vdots \\ \Omega_N \\ 1 \end{bmatrix}^T H_N \begin{bmatrix} \Omega_N^h \\ \Omega_N^{h-1} \\ \vdots \\ \Omega_N \\ 1 \end{bmatrix}. \] (3.10)
Table 1 provides the \((i, j)\)-entry of \(H_N\), where \(h := [m/2]\) and
\[
H_N(1, 1) = 0 \text{ (m odd)}, \quad H_N(1, 1) = 1 \text{ (m even)}. \tag{3.11}
\]

\[\begin{array}{c|cc}
& 3 \leq i + j \leq h + 2 & h + 2 \leq i + j \leq 2h + 2 \\
\hline
m \text{ odd} & \frac{(-C_N)^{i+j-3}}{i + j - 1} \left[ \begin{array}{c} m \\ i + j - 3 \end{array} \right] & \frac{(-C_N)^{i+j-3}}{m + 4 - (i + j)} \left[ \begin{array}{c} m \\ i + j - 3 \end{array} \right] \\
& \frac{(-C_N)^{i+j-2}}{i + j - 1} \left[ \begin{array}{c} m \\ i + j - 2 \end{array} \right] & \frac{(-C_N)^{i+j-2}}{m + 3 - (i + j)} \left[ \begin{array}{c} m \\ i + j - 2 \end{array} \right]
\end{array}\]

Moreover, \(J_n\) is a matrix satisfying
\[
\begin{bmatrix}
\Omega_n^h \\
\Omega_{n-1}^h \\
\vdots \\
\Omega_n \\
1
\end{bmatrix}
= J_n
\begin{bmatrix}
\Omega_{n-1}^h \\
\Omega_{n-1}^{h-1} \\
\vdots \\
\Omega_{n-1} \\
1
\end{bmatrix}, \quad J_n \in \mathbb{R}^{(h+1) \times (h+1)}, \tag{3.12}
\]

and can be computed by using (2.7) and (3.2) where
\[
\begin{align*}
J_n(i, j) &= 0 \quad (\text{if } j \leq i - 1), \\
J_n(i, j) &= \left( \frac{h - i + 1}{R + (\alpha_{n-1}(S_n - RS_{n-1}))} \right)^{h-j+1} \\
&\times [\beta_{n-1}(S_n - RS_{n-1})]^j \quad (\text{otherwise}). \tag{3.13}
\end{align*}
\]

### 3.2.3 Algorithm and Computational Complexity Analysis

We are now in a position to propose an algorithm to compute the \(m\)-th moment of the wealth balance distribution:

**Algorithm 1**

**Step 1:** Given \(m\), the stock process in (2.2) with probabilities \(p_l, l = 1, \ldots, L\), and the stock prices on the multinomial lattice in (2.3). Compute \(H_N\) for all \(S_N = S_N^{(k)}, k = 1, \ldots, N(L - 1) + 1\).

**Step 2:** Repeat the following step for \(n = N - 1, \ldots, 0\) until \(H_0\) is attained:
Compute $H_n = \mathbb{E} \left(J_{n+1}^T H_{n+1} J_{n+1} \mid S_n\right)$ for all $S_n = S_n^{(k)}$, $k = 1, \ldots, n(L-1) + 1$.

**Step 3)** Let

$$
\mathbb{E} \left(W_N^0 \mid S_0, \Omega_0\right) = \begin{bmatrix}
\Omega_0^0 \\
\Omega_0^{L-1} \\
\vdots \\
\Omega_0 \\
1
\end{bmatrix}^T H_0 \begin{bmatrix}
\Omega_0^0 \\
\Omega_0^{L-1} \\
\vdots \\
\Omega_0 \\
1
\end{bmatrix}.
$$

(3.14)

In the above algorithm, we calculate $H_n$ at each node for $n = N-1, \ldots, 0$, moving backward along the stock lattice to find $H_0$. Therefore the number of iterations is determined by the number of nodes on the multinomial lattice. Let $\Phi(N)$ be the total number of iterations. Since the possible number of states for $H_n$ is $n(L-1) + 1$ at each time step $n$, the total number of iterations, $\Phi(N)$, is given by

$$
\Phi(N) = \sum_{n=0}^{N-1} [n(L-1) + 1] = \frac{N((L-1)(N-1) + 2)}{2}.
$$

(3.15)

Condition (3.15) shows that the number of iterations used in Algorithm 1 is of square order with respect to the number of periods $N$ and that the total computational complexity is of polynomial order in $(N, L)$. Note that the algorithm can still be considered computationally tractable even if we increase the number of branches, $L$, since $\Phi(N)$ is a linear function of $L$.

**Remark 3.2** In general, exact computation of the wealth balance distribution is difficult. This is because the possible states for $\Omega_N$ grows exponentially in the time step $n$, i.e., e.g., $L^n$ in the $L$ state multinomial lattice case. In other words, exact computation of the VaR may involve an exponential order computation and hence is generally computationally intractable (for instance, a 20 step trinomial lattice for the stock would lead to more than $3.48 \times 10^9$ states for $W_N$). On the other hand, our proposed methodology for calculating moments is highly tractable since the moments are computed in polynomial time by propagating matrices $H_n$ ($n = 0, \ldots, N$) as given in Algorithm 1.

Although we have only explained the moment computation procedure for a European call option, the same approach can be extended to other types of options, including many exotics (such as barriers, compounds, and others) and options with time optionality (such as Americans and Bermudans). The above algorithm only requires a change in the “boundary condition” corresponding to the appropriate option type, and proper discounting to account for the time value of different wealth balance cash flows. For example, an American call option would require the additional condition that $C_n \geq (S_n - K)^+$ (in addition to $C_N = (S_N - K)^+$), where $C_n$ denotes the value of the option at time $n \in [0, N]$. Furthermore, if all cash flows are discounted to the expiration date, the wealth balance of a hedged American call option writer would be $W_N = \Omega_N - (S_N - K)^+$ if no early exercise occurred, and $W_N = R^{N-n} \left[\Omega_n - (S_n - K)^+\right]$ in the case of an early exercise at time $t_n$. 

9
4 VaR Estimation Under Moment Constraints

In this section we apply convex optimization techniques to the problem of estimating the Value-at-Risk of the wealth balance $W_N$ under moment constraints.

4.1 Convex Optimization Bounds on VaR

As a measure of risk in a distribution, for a given $\gamma$ we wish to find the value of the wealth balance $w^*$ such that the probability of exceeding this value is $\gamma$, i.e.

$$\int_{w^*}^{\infty} f_{W_N}(x) \, dx = \gamma,$$

where $f_{W_N}(x)$ is the probability density function of $W_N$. The so-called (relative) Value-at-Risk is then defined as follows [16],

$$\text{VaR}(W_N, \gamma) := \mathbb{E}(W_N) - w^* = -w^*. \quad (4.1)$$

where the last equality holds since $\mathbb{E}(W_N) = 0$ (see Subsection 3.1).

For example, if $\gamma$ is 0.95 and $\text{VaR}(W_N, \gamma = 0.95) = -w^*$, then the writer of the call option is 95% certain that he will not lose more than $w^*$ dollars in the hedge. Note that the Value-at-Risk, $w^*$, can also be thought of as the quantile of the wealth balance distribution such that the probability of a value lower than $w^*$ is $1 - \gamma$, as shown in Fig. 1.

Our approach will be to approximate the wealth balance distribution by a probability density whose first $m$ moments are the same as those for the wealth balance.
distribution. Suppose that the first $m$ moments of the wealth balance $W_N$ were obtained from Algorithm 1 as

$$ q_k := \mathbb{E} \left( W_N^k \mid S_0, \Omega_0 \right), \quad k = 1, \ldots, m. \tag{4.2} $$

and consider the set of probability density functions

$$ Q(q_1^*, \ldots, q_m^*) := \left\{ \pi(x) \geq 0 \ \mid \int_{-\infty}^{\infty} x^k \pi(x) dx = q_k^*, \quad q_0^* = 1, \quad k = 0, 1, \ldots, m \right\}. \tag{4.3} $$

We will choose a $\pi(x)$ from $Q(q_1^*, \ldots, q_m^*)$ and estimate the VaR of $W_N$ from that probability distribution. More specifically, we will use a worst case approach in the sense of VaR estimation in that we will choose the density from $Q(q_1^*, \ldots, q_m^*)$ which maximizes the possible loss under the moment constraints. This can be written as the following maximization:\(^2\)

$$ X_u(\gamma) := \max \left\{ X \ \mid \pi(x) \in Q(q_1^*, \ldots, q_m^*), \ \int_{-X}^{\infty} \pi(x) dx = \gamma \right\} \tag{4.4} $$

Note that in some optimal portfolio asset allocation literature, the worst case estimate of VaR under moment conditions is referred to as “worst case VaR” [6, 18].

Since the estimate in (4.4) is the worst possible given the moment constraints, it should be clear that the true VaR of $W_N$ is less than $X_u$. Similarly, one may bound the VaR from below by solving:

$$ X_l(\gamma) := \min \left\{ X \ \mid \pi(x) \in Q(q_1^*, \ldots, q_m^*), \ \int_{-X}^{\infty} \pi(x) dx = \gamma \right\}. \tag{4.5} $$

where the gap between $X_u(\gamma)$ and $X_l(\gamma)$ is reduced with additional moment information.

**Remark 4.1** In general, one may choose an arbitrary density function in the set $Q(q_1^*, \ldots, q_m^*)$ and use that to estimate the VaR. In particular, using a Gaussian approximation can be thought of as fitting the first two moments, while other distributions such as the Johnson [14], Pearson [17], and generalized lambda [21] allow for fitting up to four moments and provide more accurate estimates of VaR than a Gaussian. Beyond four moments, a different approach must be employed due to the lack of convenient parameterizations of densities. Standard non-parametric approaches based on cumulant expansions exist (e.g. Cornish-Fisher Expansions [17, 15]). However, they do not require that the corresponding density function be positive. Due to these difficulties, we have chosen to employ an optimization approach which not only can be implemented efficiently, but allows for the incorporation of higher order moments.

\(^2\)In (4.4), $-X$ corresponds to $w^*$.  

11
4.2 Solution of Optimizations

In the optimization (4.4), the constraints are convex with respect to \( x \). Therefore, if \( X \) were fixed, the problem of minimizing \( \gamma \) through an appropriate choice of \( \pi(x) \) would be an infinite dimensional convex optimization problem. Based on the results of Bertsimas and Popescu [3], the dual of this problem may be reduced to a semidefinite programming problem which can be solved efficiently using interior point methods.

The dual problem can be interpreted as a generalization of the well known Markov and Chebyshev bounds, and the solution to the dual problem is equal to the solution of the primal. The basic idea is as follows: The probability in the tail of a distribution can be thought of as an expectation of an indicator function, i.e.,

\[
\text{Prob}(W_N \leq x) = \mathbb{E}(1_{\{W_N \leq x\}}).
\]  

(4.6)

If another function bounds this indicator function from above, then its expectation will be an upper bound on \( \text{Prob}(W_N \leq x) \). The Markov bound uses a linear function to bound the indicator function and requires knowledge of the mean of the distribution, while the Chebyshev bound uses a quadratic and requires knowledge of the mean and variance. In general, we may bound the indicator function with a polynomial of order \( m \), in which case we need \( m \) moments of the distribution. The optimization is then to find the best \( m \)-th order polynomial which bounds the indicator function. This problem can be reformulated as a semidefinite programming problem and solved efficiently.

We can use this to solve for the worst case VaR, \( X_u(\gamma) \) by iterating on \( X \) as follows. For each given value of \( X \) we can compute the minimum corresponding \( \gamma^* \) using semidefinite programming. If \( \gamma^* \) is less than the desired \( \gamma \), we increase \( X \), otherwise, we decrease \( X \), until we eventually achieve \( X \) such that the minimum \( \gamma^* \) is the desired \( \gamma \). At that point, \( X \) will be the worst case VaR \( X_u(\gamma) \).

The iteration over \( X \) can be done more formally as a bisection algorithm. If we begin with upper and lower bounds on \( X_u(\gamma) \),

\[ X_1 \leq X_u(\gamma) \leq X_2, \]

then we can find the worst case VaR with arbitrarily small tolerance \( \epsilon > 0 \) by solving \( N \) semidefinite programming problems where \( N \) is given by:

\[ N = \left\lceil \ln \left( \frac{X_2 - X_1}{\epsilon} \right) \cdot \frac{1}{\ln 2} \right\rceil. \]

Similarly, a lower bound \( X_l(\gamma) \) can be found with arbitrarily small tolerance \( \epsilon > 0 \) by solving semidefinite programming problems. See [3] for the exact semidefinite programming conditions.

If we have additional information about the allowable set of probability densities for \( W_N \), it is possible to improve the bounds, and the problems may still be considered computationally tractable if the set of \( \pi(x) \) is characterized as a convex set. For example, we might be able to add “monotonicity constraints” or a “maximum likelihood condition” with respect to the mean value \( q_1 \) of the distribution such as

\[
\frac{d\pi}{dx} > 0 \text{ (if } x < q_1), \quad \frac{d\pi}{dx} < 0 \text{ (if } x > q_1), \quad \frac{d\pi}{dx}|_{x=q_1} = 0. \]  

(4.7)
As will be confirmed by numerical experiments in the next section, the wealth balance distribution approximately satisfies these conditions and it is fair to say that these are reasonable assumptions. Moreover, since the set of \( \pi(x) \) under the conditions in (4.7) is convex, the problem of computing VaR is still tractable. Although it appears that this problem may not be recast as a semidefinite programming problem, it is possible to apply a linear programming approach by discretizing the problem. Note that the linear programming approach is computationally tractable even though we may have more than 10,000 total variables and approximate the probability density as a probability mass function over a few thousand grid points in \( x \).

In the next section, we will illustrate our proposed methodology for calculating upper and lower bounds on the VaR.

5 Numerical Experiments

In this section, we estimate the VaR in a mean square optimal hedge for a European call option. After computing moments of the wealth balance distribution, we first apply the semidefinite programming approach to find hard upper and lower bounds on the VaR. These bounds are then improved by posing additional constraints as in (4.7).

5.1 Problem Data

We considered a discrete time market with \( t_N = 8 \) months, where \( N = 32 \) and represents approximately 4 trades every month.

The underlying stock price process was modeled on a trinomial lattice to approximate the geometric Brownian motion:

\[
dS_t = \mu S_t dt + \sigma S_t dz. \quad S_0 = 100
\]

with \( \mu = 0.15, \sigma = 0.20 \) (where \( z \) is a Brownian motion). Accordingly, the up, middle, and down probabilities and rates on the lattice (see Fig. 2) were given by:

\[
[p_1, p_2, p_3] := [1/6, 2/3, 1/6] \quad m := \exp(\nu \Delta t), \quad u := m \exp(\sigma \sqrt{3 \Delta t}), \quad d := m \exp(-\sigma \sqrt{3 \Delta t}),
\]

where \( \nu := \mu - \sigma^2/2 \) and \( \Delta t = t_N/(12 \times N) \). Finally, for the bond we chose \( R = \exp(r_f \Delta t) \) with \( r_f = 0.10 \).

We considered the problem of hedging three European call options on the stock, all with expiration at \( t_N \), and with strike prices \( K = 90, 100, 130 \), which correspond to the option being in-the-money (ITM), at-the-money (ATM), and out-of-the-money (OTM), respectively.

5.2 Computation of Moments and Bounds on VaR

Using dynamic programming, we solved the mean square optimal hedging problem with \( K = 90, 100, 130 \) to find the optimal initial price \( C_0 = \Omega_0 \) and the optimal
local hedging parameters $\alpha_k(S_k)$ and $\beta_k(S_k)$, $k = 0, \ldots, N - 1$. The optimal initial option prices were:

\begin{align*}
C_0 &= 16.9 \ (K = 90), \quad C_0 = 10.0 \ (K = 100), \quad C_0 = 1.02 \ (K = 130).
\end{align*}

Next, we computed up to the 8th moment of the wealth balance using Algorithm 1. For reference, we will explain the lattice based computation of the third moment using the propagation of the $H_n$ matrices. Based on the final stock prices on the lattice, we first computed $H_n^{(k)}$, $k = 1, \ldots, 2N + 1$ as

\begin{equation}
H_n^{(k)} = \begin{bmatrix}
0 & 1/2 & -C_n^{(k)} \\
1/2 & -C_n^{(k)} & (3/2)[C_n^{(k)}]^2 \\
-C_n^{(k)} & (3/2)[C_n^{(k)}]^2 & -3[C_n^{(k)}]^3
\end{bmatrix}
\end{equation}

where $C_n^{(k)} = (S_n^{(k)} - K)^+$, $k = 1, \ldots, 2N + 1$. Similar to the stock price lattice, the $H_n$’s can be arranged at the end nodes of the trinomial lattice as shown in the right hand side of Table 2. Next, we calculated $H_{N-1}^{(k)}$, $k = 1, \ldots, 2N - 1$, as in (3.6). For example, $H_{N-1}^{(1)}$ is given by

\begin{equation}
H_{N-1}^{(1)} = \mathbb{E} \left( J_N^T H_N J_N \left| S_{N-1}^{(1)} \right. \right) = p_1 J_N^{(1)T} H_N^{(1)} J_N^{(1)} + p_2 J_N^{(2)T} H_N^{(2)} J_N^{(2)} + p_3 J_N^{(3)T} H_N^{(3)} J_N^{(3)}
\end{equation}

where $J_N^{(l)}$ is calculated using (3.13) for each $S_N^{(l)}$, $l = 1, 2, 3$. Similarly, we computed $H_{N-1}^{(k)}$ for all $k = 1, \ldots, 2N - 1$. Continuing in this manner, working toward the left one period at a time, we arrived at $H_0$. The third moment was then computed through (3.14).

Table 3 provides numerical results, where the first 8 moments were computed for the three cases, $K = 90$ (ITM), $K = 100$ (ATM), and $K = 130$ (OTM), where the
Table 2: Stock price lattice and moment matrix lattice

\[
\begin{array}{cccccccc}
S_0 & S_1^{(1)} & S_2^{(1)} & \cdots & S_N^{(1)} & H_0^{(1)} & H_1^{(1)} & \cdots & H_N^{(1)} \\
S_1^{(2)} & S_2^{(2)} & \cdots & \cdots & S_N^{(2)} & H_0^{(2)} & H_1^{(2)} & \cdots & H_N^{(2)} \\
S_1^{(3)} & S_2^{(3)} & \cdots & \cdots & S_N^{(3)} & H_0^{(3)} & H_1^{(3)} & \cdots & H_N^{(3)} \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
S_1^{(5)} & S_2^{(5)} & \cdots & \cdots & S_N^{(5)} & H_0^{(5)} & H_1^{(5)} & \cdots & H_N^{(5)} \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
S_1^{(N+1)} & S_2^{(N+1)} & \cdots & \cdots & S_N^{(N+1)} & H_0^{(N+1)} & H_1^{(N+1)} & \cdots & H_N^{(N+1)} \\
\end{array}
\]

1st moments are omitted because they are always 0. Since MSOH was employed, the 2nd moment of the wealth balance is minimized, and no other hedging scheme can attain a wealth balance distribution with smaller variance.

Table 3: Moments obtained from Algorithm 2 for $K = 90, 100, 130$

<table>
<thead>
<tr>
<th>$K$</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>0.370</td>
<td>0.0131</td>
<td>1.09</td>
<td>0.130</td>
<td>6.72</td>
<td>0.928</td>
<td>60.5</td>
</tr>
<tr>
<td>100</td>
<td>0.844</td>
<td>0.176</td>
<td>3.39</td>
<td>1.77</td>
<td>26.5</td>
<td>20.6</td>
<td>303</td>
</tr>
<tr>
<td>130</td>
<td>0.665</td>
<td>0.0324</td>
<td>3.14</td>
<td>0.241</td>
<td>30.4</td>
<td>-3.14</td>
<td>437</td>
</tr>
</tbody>
</table>

Note that the moments of odd order are relatively small when compared to those of even order, which indicates that the wealth balance distributions are nearly symmetric. Additionally, the wealth balance distributions tend to be leptokurtic (or heavy tailed), particularly in the ITM and OTM cases. (Both skewness and Fisher kurtosis are provided in Table 4.)

Monte Carlo simulations were run to verify that our observations from the moment data were correct. Figs. 3–5 contain results of these Monte Carlo simulations, where each simulation trial was done 100,000 times. Notice that the wealth balance
Table 4: Skewness and Fisher kurtosis for $K = 90, 100, 130.$

<table>
<thead>
<tr>
<th>$K$</th>
<th>skewness</th>
<th>kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>0.0583</td>
<td>4.99</td>
</tr>
<tr>
<td>100</td>
<td>0.227</td>
<td>1.77</td>
</tr>
<tr>
<td>130</td>
<td>0.0597</td>
<td>4.09</td>
</tr>
</tbody>
</table>

distributions are leptokurtic in all cases, with the ITM and OTM cases being the most extreme. In fact, the distributions tend to become more leptokurtic as the option moves increasingly in the money and out of the money.

![Wealth balance distribution for $K = 90$ (ITM)](image)

Fig. 3: Wealth balance distribution for $K = 90$ (ITM)

Using the moments in Table 3, we computed upper bounds on VaR$(W_N, \gamma = 0.95)$ as given in Table 5, where $m$ represents the number of moments used. For $m = 8$, the following lower bounds were computed as well:

$0.130 \ (K = 90), \ 0.390 \ (K = 100), \ 0.217 \ (K = 130).$

To estimate the gap between these bounds and the true value, we computed the 5% quantiles in Figs. 3–5 from the Monte Carlo simulations, and obtained the following VaR estimates,

$0.975 \ (K = 90), \ 1.46 \ (K = 100), \ 1.34 \ (K = 130). \ \ (5.3)$

Since the number of trials was sufficiently large (indeed 100,000 trials), these numbers should be close to the true values, i.e., VaR$(W_N, \gamma = 0.95)$. A comparison
Fig. 4: Wealth balance distribution for $K = 100$ (ATM)

Fig. 5: Wealth balance distribution for $K = 130$ (OTM)
of these Monte Carlo estimates with the upper and lower bounds indicates that a significant gap does exist.

Table 5: Upper bounds for $m = 2, 4, 6, 8$

<table>
<thead>
<tr>
<th></th>
<th>$m = 2$</th>
<th>$m = 4$</th>
<th>$m = 6$</th>
<th>$m = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K = 90$</td>
<td>2.65</td>
<td>1.84</td>
<td>1.83</td>
<td>1.69</td>
</tr>
<tr>
<td>$K = 100$</td>
<td>4.01</td>
<td>2.44</td>
<td>2.35</td>
<td>2.30</td>
</tr>
<tr>
<td>$K = 130$</td>
<td>3.56</td>
<td>2.41</td>
<td>2.39</td>
<td>2.23</td>
</tr>
</tbody>
</table>

To improve the bounds, we introduced monotonicity constraints in addition to the moment conditions, and discretized the problem to apply a linear programming approach (see Section 4.2). After discretizing, the moment constraints in (4.3) become

$$\sum_{i=1}^{N_x} x_i^k p_i = q_k^a, \; \; q_0^a = 1, \; k = 0, 1, \ldots, m.$$  

where $x_i (i = 1, \ldots, N_x)$ are the possible outcomes of the discretized random variable, and $p_i (i = 1, \ldots, N_x)$ represents the corresponding discretized probability mass function. The monotonicity constraint in (4.7) is then given by

$$p_i < p_j \text{ (if } x_i < x_j < q_1^a), \; \; p_i > p_j \text{ (if } q_1^a < x_i < x_j).$$

Note that, from Figs. 3–5 this monotonicity constraint is not very restrictive since the wealth balance distributions appear to monotonically increase for $x < q_1^a = 0$ and decrease for $x > q_1^a = 0$. We set the following bounds on the discretized random variable $x$

- $x \in [-4, 4]$ ($K = 90$),
- $x \in [-4.5, 4.5]$ ($K = 100$),
- $x \in [-5, 5]$ ($K = 130$)

and chose $N_x = 1,200$. Under these conditions, we solved the upper and lower bound problems using linear programs. Figs. 6–8 show our numerical results, where the number of moments $m$ versus the corresponding upper and lower bounds are plotted for $K = 90, 100, 130$. In each figure, the upper and lower lines denote the upper and lower bounds using $m$ moments, and the line between them indicates the value obtained from the Monte Carlo simulations. As higher order moments are used, the gap between the upper bounds and the lower bounds decreases noticeably. In fact, under 8 moments, the bounds provide a reasonable approximation to the Monte Carlo VaR estimate. For example, the upper and lower bounds in Fig. 7 are “1.57” and “1.38,” respectively, as compared to “1.46” from the Monte Carlo simulation. This is a difference of only 8%. From these numerical experiments, we conclude that the bounds obtained from moment conditions can be improved by using additional information such as the monotonicity constraint, and that the problem remains computationally tractable, providing a reasonable estimate of the VaR.
Fig. 6: Upper and lower bounds on VaR, $K = 90$ (ITM)

Fig. 7: Upper and lower bounds on VaR $K = 100$ (ATM)
6 Conclusion

In this paper, we proposed an efficient methodology for computing moments of the wealth balance distribution in dynamic hedging problems. First, a mean square optimal hedging problem was employed to determine an optimal hedging policy and optimal initial portfolio value. By exploiting structure in this problem, we showed that moments of the resulting wealth balance distribution may be computed on the underlying stock lattice through the backward iteration of a matrix. Next, we demonstrated that these moments can be used in conjunction with convex optimization techniques to estimate the Value-at-Risk in the wealth balance. Finally, this methodology was applied to a numerical example where the VaR was computed for a hedged European call option on a stock modeled as a random walk on a trinomial lattice.

References


