Trading Volatility Mispricings in the Options Market

Anthony Yang, Lily Liu, Thipok Lertplakorn

Abstract

Volatility mispricings exist in the options market due to the lack of a deterministic way to compute implied volatility of the underlying assets. As a continuation of the first project, we aim to capture these mispricings using a different stock price simulation method and optimization model. Instead of using an Autoregressive Monte Carlo Simulation method, we simulate stock returns using a Multivariate Normal Distribution via Cholesky Decomposition. Similar to the first project, different options strategies are treated as individual assets and are traded three hours before the expiration on the Friday of every week. The Markowitz Optimization Model with Expected Shortfall constraint is utilized to optimize the portfolio.

1. Introduction

In this project, we continue our efforts to explore implied volatility mispricings in the options market and build an optimal portfolio of option trading strategies given a set of underlying assets. We use the same trading workflow as the first project: at 1:00PM on each Friday (i.e. three hours before the options expiration), however, we now simulate each stock price path using a multivariate normal distribution via Cholesky Decomposition. Based on the simulated stock prices and implied volatilities, we calculate the expected payoff at expiration for each options trading strategy and trade on optimal portfolio of strategies which maximizes the returns. Different option spreads are treated as different assets and the Markowitz Model with Expected Shortfall is applied to determine the optimal portfolio.

There are two major changes compared to the first project. Firstly, we decided to incorporate Expected Shortfall, also known as Conditional Value-at-Risk (CVaR), into the basic Markowitz Mean-Variance Optimization Model. For general distributions, CVaR, though similar to VaR measure of risk, is both sub-additive and convex, and also provides a coherent sense of risk. In addition, CVaR accounts for true outliers while VaR does not. At a 95% confidence level, the Expected Shortfall gives the expected value of an investment in the worst 5% of scenarios. In our project, we define a scenario to be the simulated set of six different, but correlated stock price paths. For example, an example scenario is demonstrated in Figure 1.

Email addresses: `axy2@cornell.edu` (Anthony Yang), `xl379@cornell.edu` (Lily Liu), `tl424@cornell.edu` (Thipok Lertplakorn)

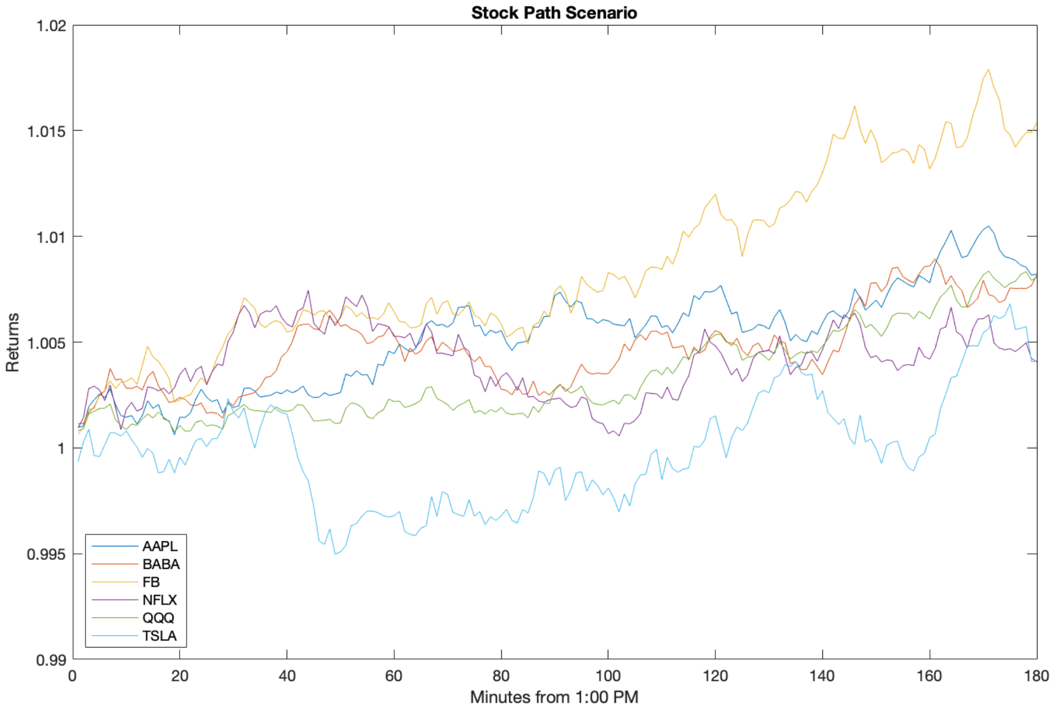

Figure 1: A scenario of six correlated stock prices

Secondly, instead of using an Autoregressive Monte Carlo method to simulate log-returns, we perform Monte Carlo simulation on a multivariate normal distribution via Cholesky Decomposition to simulate each stock price path. The main reason for this change is to accommodate the introduction scenarios in our expected shortfall calculations. In order to reduce the complexity of the model, we choose to ignore the autocorrelation in each stock's log-returns and assume that the log-returns are normally distributed – an assumption that is consistent with the Geometric Brownian Motion model.

Additionally, we improved our simulation by increasing the size of the dataset from 6 weeks to 13 weeks and maintaining the granularity of the data to be globally by the minute.

The report is organized in the following manner:

- Section 2: Data collection and cleaning, including historical stock and option prices at minute granularity
- Section 3: Assumptions of the project
- Section 4: Monte Carlo simulation using multivariate normal distribution via Cholesky Decomposition
  - To determine the expected payoff of the options, we simulate multiple paths of the underlying asset prices 3 hours before the option expiration (i.e. 1:00PM to

4:00PM) on each Friday

- Section 5: Application of Markowitz Optimization Model with Expected Shortfall and model validation
- Section 6: Testing and results of the optimal portfolio, an analysis on the strategy, and comparison to Mean-Variance Optimization portfolio
- Section 7: Conclusion

2. Data Collection and Cleaning

Following the first project, we collected more data of the following 6 stocks and their respective options chain: Apple (AAPL), Alibaba (BABA), Facebook (FB), Netflix (NFLX), NASDAQ Index (QQQ), Tesla (TSLA). The tickers chosen were essentially the most popular, and thus, the most liquid. The method of choosing the strike prices for each option and the usage of a Bloomberg Python API were discussed in the previous report. We have collected 13 weeks of data in total.

One of the biggest issues with the data gathered from Bloomberg was the inconsistencies in granularity. Because the Bloomberg data's granularity is dependent on the frequency of trade and scales with liquidity, there were often missing data for particular minutes of the day or even for hours if the option was not traded during that time period. However, having consistent formatting and data length is a key requirement of constructing the covariance matrix of the various options. As a result, we have to impute the missing values by taking the last known price and applying it for the missing time period, assuming that Bloomberg only updates its data when the price changes. While we did tackle this issue somewhat in the last project, there were a couple of edge cases that we failed to consider, which reduced our total data pool. For example, our previous model did not account for the cases where data for the entire days were missing, and would thus, omit those days completely in the final output. To be more specific, if there was no data for the whole day (i.e. the option was not traded at all on that specific day), we needed to add a "dummy price" taken from another day.

Additionally, since we have changed the stock price simulation method, we require that the numbers of prices at minute-granularity match across all 6 stocks. Even though stocks are more liquid compared to options, there are still several cases of missing data and this creates a mismatch in the number of prices among the sets. Therefore, we added an additional part of cleaning to the stock data. We applied the same method of missing value imputation as described above for the options prices.

3. Assumptions

For simplicity and clarity, we have made the following assumptions:

1. Options are only exercised at expiration (i.e. American options treated as European options)

2. Options are liquid enough so they can be traded at any time
3. Shorting is an option
4. There is negligible autocorrelation among each stock's minute log-returns
5. Log-returns are normally distributed
6. No commission fee
7. Model allows a fraction of shares to be traded

4. Stock Price Simulation Model

The general framework of the price simulation model stayed relatively similar to Project 1, however, the simulation model employed in this project is markedly different. To briefly summarize the model framework, we perform Monte Carlo simulations on a multivariate normal distribution of the six stocks' minute log-returns in order to obtain the predicted price paths and implied volatility, allowing us to calculate the expected payoff of each options trading strategy at expiration. Some key assumptions include omitting stock returns data during the first 2 $\frac{1}{2}$ hours of each trading day when fitting the model due to that time period's inconsistent and high volatility.

However, in order to account for Expected Shortfall, we need to calculate the shortfall of each possible scenario. Unfortunately, our original AR(5) model simulated 10,000 paths of each of the 6 stocks, implying that there would be a total of $10{,}000^6$ scenario permutations. The time complexity of sorting algorithms are generally, under big O notation, "*n log(n)*" on average and "n^2" at worst. Therefore, sorting $10{,}000^6$ scenarios would take an unreasonable amount of time and computing power. In addition, another issue with our original simulation was that our model simulated each stock's path independently from each other. In reality, stocks rarely act this way since there are general market trends and there exists varying correlations between stocks. Two tech stocks, for example, would be much more correlated than a tech stock and an agricultural stock.

Our proposed change to the model is to simulate all six stock paths for a given scenario as one correlated entity. By assuming that minute log-returns are normally distributed with 0 autocorrelation, which matches the same assumptions under a Geometric Brownian Motion model, we generated a Multivariate normal distribution using Cholesky Decomposition:

$$\textit{Let } R_1 \sim N(\mu_1, \sigma_1^2), ..., R_6 \sim N(\mu_6, \sigma_6^2),\ \boldsymbol{R} = (R_1, ..., R_6)^T,\ \boldsymbol{\mu} = (\mu_1, ..., \mu_6)^T,\ \textit{then}$$
$$\boldsymbol{\mu} + Chol(\boldsymbol{\Sigma})\boldsymbol{R} \sim N_6(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where μ_i is the mean log-return for stock i and σ_i^2 is stock i's log-return variance in a given week.

After we obtain the 10,000 scenarios that each of the 6 stocks could move, we then input these stock price scenarios into our option spread payoff functions, obtaining 10,000 scenarios of how each option spread (i.e. straddle, condor, and put spread) could potentially pay off. After subtracting the position costs from these payoff scenarios, we finally obtain 10,000 profit/loss scenarios for each option spread for a specific week.

Figure 2 shows four examples of three-hour simulated stock price scenarios, while the flowchart in Figure 3 summarizes the processes required to obtain the final datasets and scenarios.

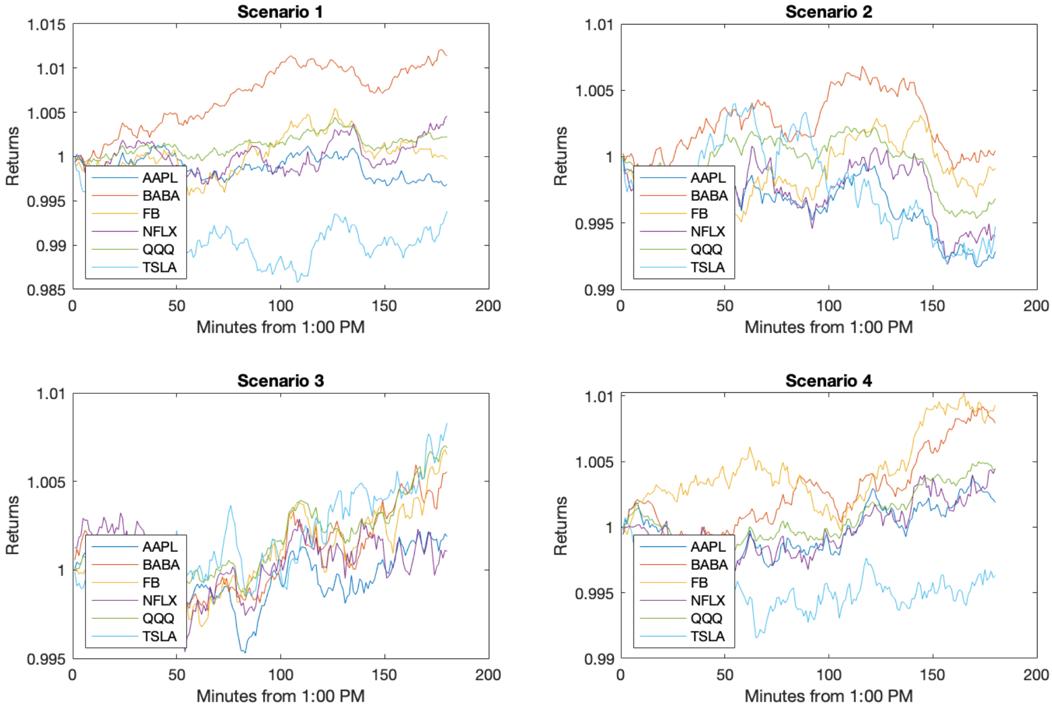

Figure 2: Four different stock price simulation scenarios

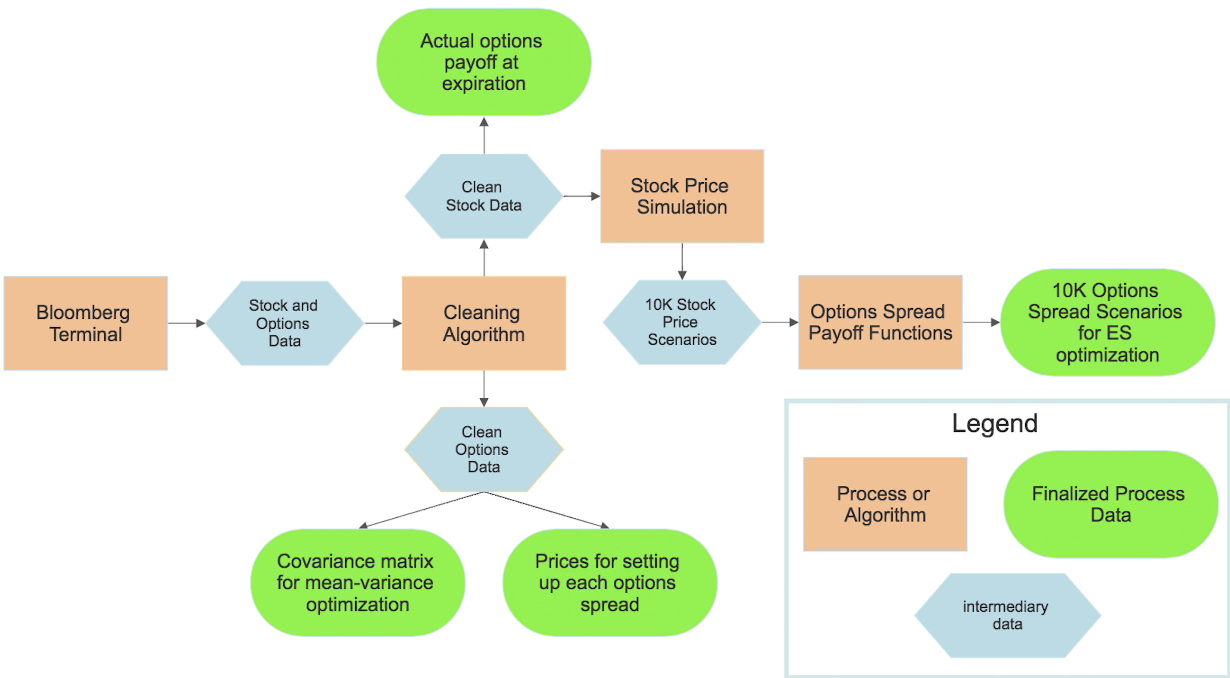

Figure 3: Workflow Summary

5. Optimization Model

In our previous project, we utilized the Mean-Variance Optimization model in order to construct a portfolio of spreads with the highest expected value, for a given portfolio risk. As a recap, we sought to maximize the expected return, such that the standard deviation of our portfolio is no more than 10% of our net assets. We also limited both the net credit and net debit of positions to be no more than 5% of our net assets and restricted the maximum number of shares of each viable strategy to 300 in order to account for realistic market liquidity. Below, is the old model's linear program:

$$
\begin{aligned}
&\max x^T\mu \\
&\text{s.t.} \\
&x^T V x \leq (0.05P)^2 \\
&x^T c \leq 0.05P \\
&x^T c \geq -0.05P \\
&x \geq -100 \\
&x \leq 100
\end{aligned}
$$

where V is the covariance matrix of each options strategy, P is the current portfolio value, c is the vector of costs to set up each strategy, μ is the expected return vector, and x is the portfolio weights.

In this project, we increased the complexity of our optimization model by introducing expected shortfall, or "conditional value at risk" (CVaR), as an additional constraint. However, the difficulty in computing expected shortfall lies when a portfolio x, given a user-chosen "confidence level" α, is chosen to minimize $ES_\alpha(x)$. The traditional definition that $ES_\alpha(x)$ is the average loss among the worst α of the scenarios falls short because that definition depends on the samples being ordered according to the loss incurred, but the loss (and hence the ordering) depends on the portfolio x. However, Rockafellar and Uryasev (2000) demonstrated that a linear programming constraint can be introduced in an optimization model the Expected Shortfall risk measure which avoids ranking the scenarios:

$$\text{ES}_\alpha(x) = \min_{\ell}\left(\ell + \frac{1}{1-\alpha}\sum_{\omega\in\Omega} p(\omega)\max\{\text{loss}(x,\omega) - \ell, 0\}\right)$$

Crucially, if the loss function $loss(x,\omega)$, for a given scenario ω and portfolio x , is convex, then Rockafellar and Uryasev's formula for expected shortfall is also convex in the variables (x,ℓ) . This cannot be understated since convex equations can be used within linear programming and thus, the ES formula can be used as a constraint under CVX. The updated linear program is as follows:

$$
\begin{gathered}
\max x^T\mu \\
\text{s.t. } x^T c \leq 0.05P \\
x^T c \geq -0.05P \\
x \geq -100 \\
x \leq 100 \\
\ell + \tfrac{1}{1-\alpha}\textstyle\sum_{\omega\in\Omega} p(\omega)\max\left\{-R(\omega)^T x - \ell, 0\right\} \leq \beta P
\end{gathered}
$$

where P is the current portfolio value, c is the vector of costs to set up each strategy, μ is the expected return vector, x is the portfolio weights, α is the desired confidence level, ω is a scenario in the space of all possible scenarios Ω, $p(\omega)$ is the probability of scenario ω happening, and $R(\omega)$ is the expected returns of scenario ω.

Finally, while typical optimization models utilize percentages and fractions as the preferred metrics, our model uses dollar values to better capture the objective of the project and for ease of interpretation.

5.1. Model Validation

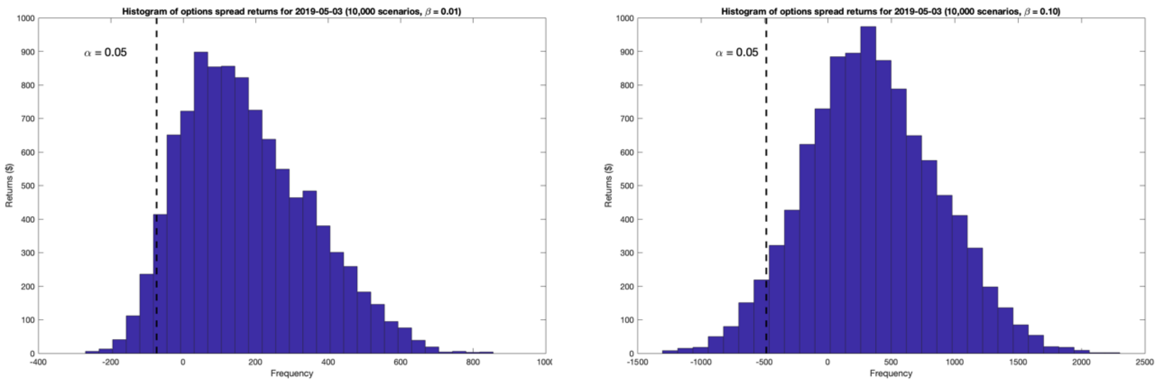

Figure 4: Spread returns for 10,000 scenarios under our optimal portfolio, β= 0.01 and β=0.10

Week (Before Trading)	1	2	3	4	5	6	7	8	9	10	11	12	13
Expected Shortfall	$100.00	$101.63	$90.87	$102.11	$70.89	$45.92	$63.38	$107.38	$108.65	$96.77	$49.39	$111.12	$114.73
Portfolio Net Worth	$10,000	$10,163	$10,117	$10,211	$10,334	$10,523	$10,574	$10,738	$10,890	$10,838	$11,059	$11,113	$11,473
E.S./Portfolio Net Worth	0.010	0.010	0.009	0.010	0.007	0.004	0.006	0.010	0.010	0.009	0.004	0.010	0.010

Figure 5: Preliminary testing results for β=0.01

In order to check for the validity of the Expected Shortfall constraint in our optimization model, we performed preliminary testing on the results of the optimization models with simplified constraints. By setting our β to be a very small value, we expect the portfolio allocation to be conservative, with a very low possibility of large losses. On the other hand, when we set our β to be very large, we expect an aggressive portfolio allocation with a higher expected return, but also higher possibilities for large losses.

The result of our test confirms the validity of our optimization model. When we set β to be 0.01 (expected shortfall no larger than 1% of current portfolio liquid worth), we obtain the following results as shown in Figures 5 and 4(left).

From Figure 4(left), we can see that at no point in the trading simulation is our expected shortfall larger than 1% of our portfolio net worth.

On the other hand, setting $\beta = 0.10$ yields the results as shown in Figures 6 and 4(right).

As seen in Figures 6 and 4(right), we observed that the Expected Shortfall for each week of trading is much larger than the previous case. Additionally, the histogram for our profit and loss in the 10,000 scenarios has a markedly higher variance and expected value.

Week (Before Trading)	1	2	3	4	5	6	7	8	9	10	11	12	13
Expected Shortfall	$567.20	$903.80	$586.80	$491.30	$662.90	$801.50	$1,023.80	$553.80	$515.50	$651.00	$601.10	$418.10	$677.50
Portfolio Net Worth	$10,000	$10,974	$11,170	$11,363	$11,672	$12,245	$12,621	$13,737	$14,025	$14,036	$14,388	$14,742	$15,499
E.S./Portfolio Net Worth	0.057	0.082	0.053	0.043	0.057	0.065	0.081	0.040	0.037	0.046	0.042	0.028	0.044

Figure 6: Preliminary testing results for β=0.10

The results obtained from our preliminary testing suggest that the optimization model is functioning properly, as it produces results that are in line with our expectations.

6. Testing and Results

6.1. Markowitz Mean-Expected Shortfall Optimization

After we ran the trading simulation on the full amount of data (13 weeks) with the optimization constraints described in Section 5, $\alpha = 0.95$ and $\beta = 0.05$, we obtain the following evolution of portfolio net worth as seen in Figure 7 and Figure 8.

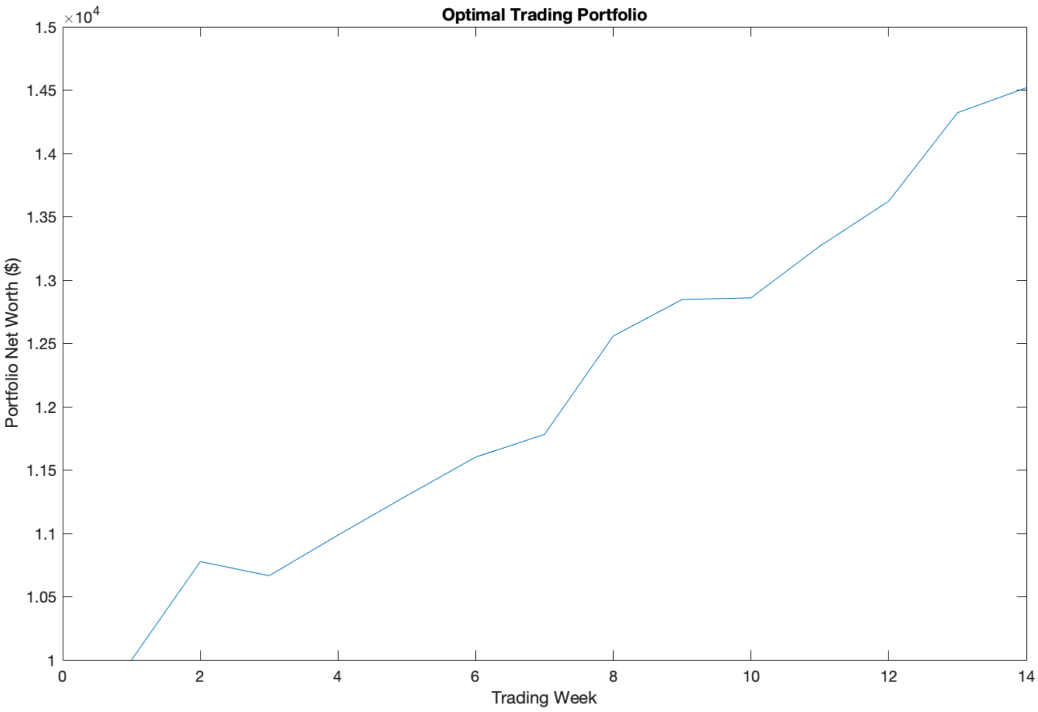

Figure 7: Portfolio Evolution, $\alpha = 0.95$, $\beta = 0.05$

From Figure 7 and Figure 8, we see that our portfolio grew 45.19% over thirteen weeks. During the period, we only had one week where the portfolio had a negative return, yielding -\$112. The $\alpha = 0.95$ Expected Shortfall for each week remains less than 0.05 times the portfolio net worth throughout the 13 weeks, as expected.

Week (Before Trading)	1	2	3	4	5	6	7	8	9	10	11	12	13
Expected Shortfall	$424.59	$445.07	$520.70	$495.50	$470.36	$564.53	$589.07	$553.78	$496.65	$556.05	$597.99	$381.64	$677.55
Portfolio Net Worth	$10,000	$10,777	$10,665	$10,986	$11,298	$11,602	$11,781	$12,559	$12,847	$12,860	$13,268	$13,623	$14,322
E.S./Portfolio Net Worth	0.042	0.041	0.049	0.045	0.042	0.049	0.050	0.044	0.039	0.043	0.045	0.028	0.047

Figure 8: Tabulated Portfolio Evolution, $\alpha = 0.95,\ \beta = 0.05$

Spread/Week	Position Size												
	1	2	3	4	5	6	7	8	9	10	11	12	13
AAPL Straddle	99.94	-99.95	100.00	-100.00	-99.95	-100.00	-2.94	100.00	-100.00	-49.11	100.00	-99.99	-100.00
AAPL Condor	99.99	-58.88	100.00	-100.00	-99.99	-100.00	-100.00	100.00	100.00	99.98	-100.00	-100.00	100.00
AAPL Put Spread	-99.96	-77.09	-100.00	100.00	-99.38	-100.00	100.00	100.00	-100.00	98.51	-100.00	100.00	100.00
BABA Straddle	-99.98	-99.99	100.00	-100.00	-86.88	100.00	-100.00	-100.00	100.00	-99.99	100.00	-100.00	-100.00
BABA Condor	99.99	-99.98	-90.66	-100.00	-100.00	100.00	54.35	100.00	-100.00	-100.00	100.00	-100.00	100.00
BABA Put Spread	-99.99	99.95	-100.00	24.75	99.99	100.00	-100.00	-100.00	100.00	-99.99	-100.00	-99.17	-100.00
FB Straddle	99.99	99.99	100.00	100.00	99.99	-100.00	-100.00	100.00	100.00	-99.98	92.42	-100.00	100.00
FB Condor	99.99	-99.96	-100.00	100.00	100.00	-100.00	-100.00	100.00	100.00	-100.00	-100.00	100.00	100.00
FB Put Spread	-99.38	99.99	100.00	100.00	-100.00	100.00	100.00	-100.00	99.98	52.16	-100.00	-100.00	100.00
NFLX Straddle	-75.54	-58.11	-100.00	100.00	-33.92	95.90	63.86	100.00	-83.52	-100.00	100.00	100.00	100.00
NFLX Condor	78.37	-89.80	100.00	100.00	22.14	100.00	-100.00	100.00	100.00	-40.37	100.00	100.00	100.00
NFLX Put Spread	-99.97	-99.93	99.99	-100.00	-99.99	25.55	100.00	-100.00	-100.00	100.00	-100.00	-100.00	-100.00
QQQ Straddle	99.98	-99.98	-99.70	-100.00	-99.99	100.00	-100.00	100.00	-100.00	-100.00	-100.00	-100.00	100.00
QQQ Condor	98.79	-99.99	100.00	-100.00	-99.90	-100.00	-100.00	100.00	-100.00	-100.00	-100.00	-100.00	100.00
QQQ Put Spread	-99.99	93.75	100.00	-100.00	-99.99	100.00	-100.00	-100.00	-100.00	-100.00	-100.00	-100.00	-100.00
TSLA Straddle	-63.13	-30.44	100.00	-100.00	-56.71	-100.00	-46.62	100.00	-99.96	-92.16	-99.93	-100.00	-100.00
TSLA Condor	13.87	-85.77	-100.00	-100.00	99.98	-100.00	-100.00	-100.00	100.00	-99.98	-100.00	-4.08	-100.00
TSLA Put Spread	-34.21	79.39	100.00	-100.00	99.45	-99.99	76.23	-100.00	-100.00	87.42	100.00	100.00	100.00

Figure 9: Trade Log

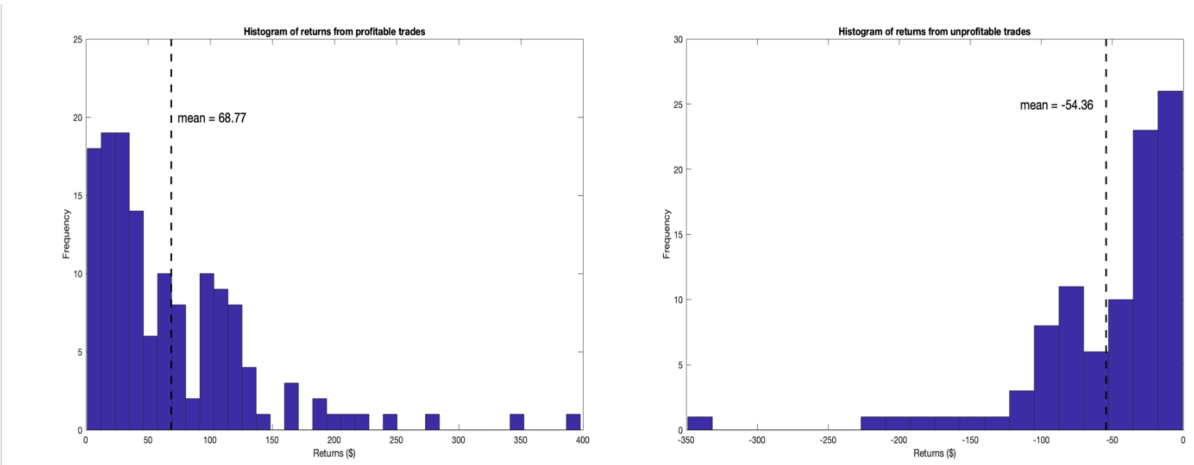

Figure 10: Histogram of returns from winning trades (Left) and returns from losing trades (right)

In addition, Figure 9 is a trade log of all trades that are executed during the thirteen-week period.

From Figure 9, our optimization algorithm executed 30 long straddle positions and 48 short straddle positions, 36 long condor positions and 42 short condor positions, and 34 long put spread positions and 44 short put spread positions.

Via the structure of our three spreads (i.e. straddle, condor, and put spread), the long positions correspond to long vega (short theta) trades and the short positions correspond short vega (long theta) trades. The bias towards short vega trades is consistent with a landmark research by Cowal and Shumway in 2001[1], which concludes that the market tends to overprice implied volatility as investors often buy options to hedge against volatility risk.

Further analysis of this data reveals that 59.83% of the trades we make are profitable. A crucial detail that we observed is that, on average, the magnitude of returns that we earn from our profitable trades is larger than the magnitude of returns from our losing trades as shown in Figure 10.

A noteworthy observation from Figure 9 is that the algorithm tends to make trades that are either long or short the full 100 contracts. This suggests that the maximum amount of contracts constraint is often bounding the optimization problem. As an experiment, if we lift the maximum amount of contracts per spread from 100 to 300, we achieve the result for our weekly Expected Shortfall levels and trade log as shown in Figure 11.

Week (Before Trading)	1	2	3	4	5	6	7	8	9	10	11	12	13
Expected Shortfall	$500.00	$527.90	$418.67	$532.98	$502.06	$568.72	$580.80	$637.97	$677.29	$263.18	$695.33	$554.91	$677.15
Portfolio Net Worth	$10,000	$10,558	$10,256	$10,660	$10,650	$11,375	$11,616	$12,760	$13,556	$13,150	$13,907	$14,252	$15,698
E.S./Portfolio Net Worth	0.050	0.050	0.041	0.050	0.047	0.050	0.050	0.050	0.050	0.020	0.050	0.039	0.043

Figure 11: Tabulated Portfolio Evolution (300 contracts limit)

We can see that, in Figure 12, our trading model executes trades the maximum number of contracts less often than in the previous case. Furthermore, the Expected Shortfall as a fraction of the portfolio for each week is often equal to the beta value of 0.05.

6.2. Comparison to Mean-Variance Optimization portfolio

Using the Mean-Variance optimization trading model from Project 1, we obtain the evolution for our portfolio over 13 weeks and the corresponding expected $\alpha = 0.95$ expected shortfall as seen in Figure 13.

In comparison to the portfolio in Part 6.1, the Mean-Variance Optimization strategy gives us a portfolio with higher absolute return over 13 weeks. However, we can see from Figure 14 that the Expected Shortfall levels often exceed 0.05 times the portfolio net worth, and at the highest point, reach 0.089 times the portfolio. This is a considerably higher amount of risk than the portfolio that we obtained in Part 6.1. An important fact to note is that while comparing the two strategies between Mean-Variance and Expected Shortfall Optimization, we used the same stock price simulation model (Multivariate Normal Distribution via Cholesky Decomposition).

[1](https://onlinelibrary.wiley.com/doi/abs/10.1111/0022-1082.00352)

Spread/Week	Position Size												
	1	2	3	4	5	6	7	8	9	10	11	12	13
AAPL Straddle	243.84	196.20	297.53	-283.25	-57.84	-19.62	187.65	300.00	-128.35	296.97	206.73	-106.91	-268.59
AAPL Condor	300.00	-300.00	299.98	-300.00	-299.99	-300.00	-300.00	300.00	300.00	-74.02	-300.00	-299.97	299.996
AAPL Put Spread	-122.10	80.46	-299.85	300.00	71.82	-300.00	300.00	300.00	-300.00	253.94	201.64	299.89	299.95
BABA Straddle	-300.00	-275.54	-69.42	-2.63	-15.09	300.00	-115.86	-262.12	105.32	-280.06	300.00	-279.98	-27.062
BABA Condor	300.00	-300.00	90.48	-300.00	-300.00	300.00	300.00	300.00	-300.00	-68.03	300.00	-299.89	299.993
BABA Put Spread	-286.14	-8.92	-224.60	-44.37	299.99	220.38	-300.00	-300.00	300.00	-299.32	-62.98	220.31	-299.85
FB Straddle	300.00	300.00	198.51	300.00	299.99	-300.00	81.71	300.00	300.00	-89.24	-256.39	9.39	231.188
FB Condor	300.00	-297.42	-267.37	300.00	300.00	-300.00	-174.18	300.00	300.00	10.26	-300.00	299.96	299.994
FB Put Spread	182.91	244.46	299.85	300.00	-300.00	-4.77	300.00	227.76	238.42	126.94	-126.84	-285.12	298.769
NFLX Straddle	-63.19	-70.32	-208.52	19.09	-143.21	-101.09	29.39	-82.49	12.61	-180.62	300.00	277.30	-154.76
NFLX Condor	-105.10	14.97	278.53	34.29	298.24	225.29	66.53	300.00	300.00	270.93	300.00	-78.97	299.875
NFLX Put Spread	-133.43	-169.88	-188.24	-30.71	-299.58	-200.51	300.00	-264.54	-154.62	16.99	-300.00	-97.50	-260.7
QQQ Straddle	111.55	-300.00	-299.89	-300.00	-299.98	300.00	-300.00	-294.78	-300.00	-299.44	-300.00	-299.96	299.951
QQQ Condor	-300.00	-300.00	299.53	-202.27	-217.81	-300.00	-300.00	300.00	-300.00	-297.49	-300.00	-260.12	299.979
QQQ Put Spread	-300.00	-300.00	299.61	-300.00	-299.99	300.00	-300.00	-300.00	-300.00	-298.74	-300.00	-299.96	-299.91
TSLA Straddle	-155.78	-59.76	-130.65	-1.91	-106.68	-33.43	-144.60	76.81	-145.07	-135.36	-256.79	-292.16	140.555
TSLA Condor	154.12	50.88	121.99	-190.11	295.51	-139.72	154.13	-279.27	300.00	163.57	197.14	299.22	-279.24
TSLA Put Spread	-27.73	71.49	299.91	-130.94	27.96	-79.94	-120.03	-300.00	94.52	148.52	300.00	284.46	299.808

Figure 12: Trade log (300 contracts limit)

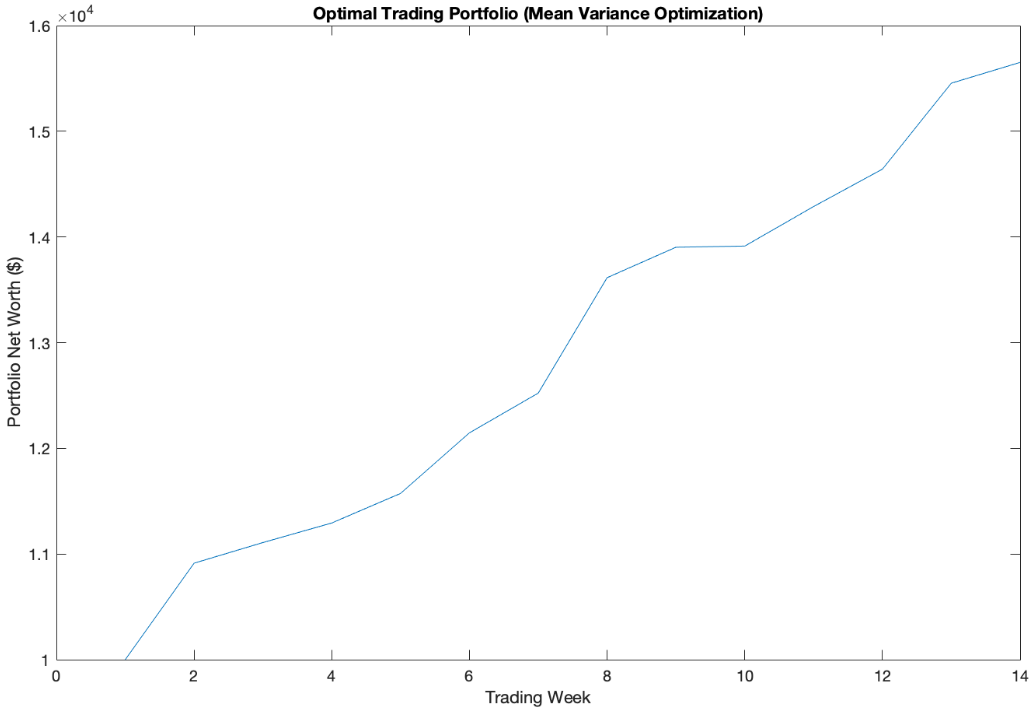

Figure 13: Portfolio Evolution (Mean Variance Optimization)

Week (Before Trading)	1	2	3	4	5	6	7	8	9	10	11	12	13
Expected Shortfall	$530.10	$903.80	$592.10	$455.90	$662.90	$801.50	$1,114.10	$553.80	$515.50	$639.20	$601.10	$461.30	$677.50
Portfolio Net Worth	$10,000	$10,913	$11,108	$11,293	$11,573	$12,146	$12,522	$13,615	$13,903	$13,914	$14,288	$14,642	$15,455
E.S./Portfolio Net Worth	0.053	0.083	0.053	0.040	0.057	0.066	0.089	0.041	0.037	0.046	0.042	0.032	0.044

Figure 14: Tabulated Portfolio Evolution (Mean Variance Optimization)

Lastly, the portfolio obtained from Part 6.1 has a Sharpe Ratio of 3.1916 while the portfolio obtained from Mean-Variance Optimization has a lower Sharpe Ratio of 3.1257, implying that the Expected Shortfall optimization trading is the more attractive strategy.

7. Conclusion

In this project, we continued our work from our first project, focusing on implied volatility mispricings in the options market by building an optimal portfolio of option trading strategies using the Markowitz Mean-Expected Shortfall Optimization. Under reasonable assumptions, the objective of the model is still to maximize the expected returns subject to given risks.

According to the testing results obtained from running the trading strategy over the entire 13-weeks data, we observe that our portfolio leads to a growth in net assets with an even lower volatility comparatively to the growth associated with the Markowitz Mean-Variance Optimization strategy.

However, our model suffers from three limitations. First, the lack of computing power restricts the number of scenarios to around 10,000. Second, while more than our last project, 13 weeks of data is still too little to draw any firm conclusion that Mean-Expected Shortfall Optimization beats out Mean-Variance Optimization or even if our portfolio will outperform the naive strategies under all market conditions. However, we are confident enough to conclude that we do find it a viable trading strategy in general. Third, assuming that our log normal returns are not autocorrelated may not be an accurate assumption of reality.

For potential future improvements, firstly, we plan on utilizing a GPU to allow us to run greater computationally expensive programs in order to increase the total amount of simulated paths. Furthermore, we can increase the size of the data by continuously gathering more prices through Bloomberg. In addition, we can extend our potential assets to include new spreads, different stocks and even commodities. Finally, we can run simulations and testing on user input parameters, such as alpha and beta, in order to construct a more robust model by finding their optimal value assignments.

Author biography

Anthony Yang Cornell Undergraduate: Computer Science Major, Operations Research Minor

Lily Liu Cornell Undergraduate: Computer Science and Operations Research Major

Thipok Lertplakorn Cornell Undergraduate: Operations Research Major