



Optimal Options Investment Strategy Final Report



Prepared for: Dr. K. B. Laskey

Prepared by:

Tony Chen

Ehsan Esmailzadeh

Ali Jarvandi

Ning Lin

Ryan J. O'Neil

Sponsor: Dr. K. C. Chang

*SEOR Department, George Mason University
Spring 2010*

Executive Summary

The optimal options investment strategy group uses mathematical modeling to find high yield strategies with low risk of catastrophic loss. Most fund managers trade on speculation or intuition based on market data rather than search for optimal strategies using a trading model. A mathematical model provides a spectrum of strategies which produce higher returns, lower risk, or both.

Investors use a very large set of options strategies combining multiple options and positions. To finish the model in the allotted time, the scope is narrowed to a single strategy with limited parameters. The strategy we optimize is the short strangle taking into account these parameters: trade date, put strike price, call strike price, stop-loss, maximum acceptable volatility, and optimal fractional allocation.

The results of the model output show that over short periods of time there are clear optimal solutions, though the same strategy will probably not remain optimal for long periods of time. Analysis of 2004-2006 showed substantially different optimal solutions than 2007-2009, therefore we recommend continuously using recent data to find current optimal investment strategies.

The optimal strategy for 2007-2009 uses call strike +5, put strike -15, stop-loss 20, sold 42 days before expiration, and is fully invested. The return on investment is over 700 times the initial investment over 3 years, or 36 trades. Sensitivity analysis shows that small changes to input parameters can significantly diminish returns but will still result in strategies with above average returns.

We have identified assumptions and limitations in the model which result in overstated returns. Perfect execution of stop-loss is the greatest factor causing inflated model output. There are many areas of future work which can make the model results more realistic and accurate.

Table of Contents

1	Introduction.....	1
1.1	Problem Statement.....	1
1.2	Statement of Need.....	1
1.3	Business Case.....	2
1.4	Background	2
1.4.1	Strangle	3
1.4.2	Payoffs	3
1.4.3	Stop-loss.....	5
1.4.4	Convergence of Options Premium and Payoffs at Expiration.....	6
1.4.5	Margin Account.....	6
1.4.6	Fractional investment.....	6
1.5	Volatility Smile.....	8
1.6	Previous Work.....	8
2	Objectives and Scope	9
2.1	Strangle Strategy Scope.....	9
2.1.1	Option Parameters	9
2.1.2	Modifications.....	9
2.1.3	Data Limitations	9
3	Technical Approach	10
3.1	Assumptions.....	10
3.2	Data.....	11
4	Model and Architecture.....	13
4.1	Preparing Data for Model.....	13
4.2	Modeling Strategies	13
4.3	Modeling the Optimal Fractional Investment Allocation.....	14
4.4	Graphical User Interface.....	14
5	Results.....	19
5.1	Days before Expiration.....	19
5.2	Stop-loss	22
5.3	Strike Prices.....	23
5.4	Volatility Driven Decision Making	26
5.5	Optimal Strategy	27
5.5.1	Sensitivity Analysis.....	27
5.6	Optimal Fractional Investment Allocation and Risk of Ruin.....	29
6	Evaluation	31
7	Recommendation	32
8	Future work	32
9	Bibliography	34
10	Appendices.....	i
10.1	Appendix 1: Requirements.....	i
10.1.1	General Requirements.....	i

10.1.2	GUI Model Requirements.....	i
10.1.3	Data Requirements	iii
10.2	Appendix 2: Project Plan.....	iv
10.2.1	Introduction	iv
10.2.2	Work Breakdown Structure (WBS)	iv
10.2.3	Individual Tasks	v
10.2.4	Project Budgeting and Earned Value Management	vii
10.2.5	Project Development Schedule (Gantt chart)	ix
10.3	Appendix 3: Strangles with Adjustments	xii
10.4	Appendix 4: Algorithms	xiv
10.4.1	Trade and Forget	xiv
10.4.2	Strangle with Stop-Loss.....	xiv
10.4.3	Strangle with Volatility Driven Decision Making	xv
10.5	Appendix 5: Pseudo Code	xv
10.5.1	Optimal Fractional Investment	xv
10.5.2	Modeling Risk of Ruin.....	xvi
10.6	Appendix 6: Sample Validation File	xvii
10.7	Appendix 7: Example of Volatility Smile	xix
10.8	Appendix 8: Optimal Fraction and Risk of Ruin Formulas	xx
10.8.1	Kelly Formula.....	xx
10.8.2	Vince Formula	xxii
10.9	Appendix 9: Final TWR & Risk of Ruin Example Code	xxiii
	TWR & Risk of Ruin Example Illustration	xxiv
10.10	Appendix 10: Investment Model Code.....	xxv
10.11	Appendix 12: Output Data Summary	xxxii

1 Introduction

1.1 Problem Statement

Investment risk and return can be improved through operations research modeling to optimize trading strategies. Currently, top option investors make trades based on their ability to predict the state of the market rather than mathematical modeling. A traditional stock trade is also known as a spot market trade. It requires immediate investment of the current value of a stock. Alternatively, option trading is a higher risk and higher reward investment than traditional stock, as it requires significantly less cash on hand. Investors can leverage their equity multiple times using options in the hopes of attaining higher returns.

There are many tools to analyze stock portfolios, but not many that analyze options trading. Professional investment funds sell investment strategies to customers based on past performance. Picking a winning strategy is difficult, and it becomes even more difficult when considering that all investors may not necessarily have the same risk tolerances. Young investors may be willing to take higher risks for a chance at higher payoffs because they expect to have many more productive years to make up for losses. On the other hand, older investors may not be willing to take higher risks as they need their assets for retirement. Therefore there may be several top strategies depending on an investor's objectives for return and risk. Mathematical modeling simulates multiple outcomes based on varying constraints providing scientific justification for investment choices in lieu of subjective feelings about the market.

1.2 Statement of Need

There is a need for a detailed analysis of option investment strategies to find optimal choices for investors with different goals. Strategies must be robust so that changes in the market do not have a large effect on the payoff or risk of ruin. There is also a need for a computer application that can display equity curves and performance of past options so that investors can test recommended strategies.

1.3 Business Case

Investment fund managers control the assets of many clients. To keep customers they must provide satisfactory services. Their goals should be to make decisions in accordance with their customers' financial interests. Our model and tools can aid fund managers to quickly assimilate information about the current options market conditions and act before the market changes. A fund manager can use our model to find the optimal strategy over a specified period of recent history. Once top strategies are found, the fund manager can apply fractional investment to match returns and risk of ruin to customer demand.

1.4 Background

A futures contract is a contractual agreement between two parties to trade an underlying asset at some specified date in the future at some specified price. An options contract is a variation of a futures contract where the contract buyer chooses whether or not the contract will be exercised at expiration. The contract seller receives a premium in return. Options are standardized by strike price and expiration date to facilitate trading. Otherwise finding a willing partner could be difficult (Hull). Definition and explanations of options parameters are listed below:

- **Position** – describes the parties entering the contract. The long position is the options buyer and the short position is the options seller or writer. It is important to distinguish position since buyers and sellers have different options payoffs.
- **Expiration Date** – date at which the options buyer decides whether or not to exercise the option.
- **Strike Price** – price at which the underlying asset is traded on the expiration date. Options buyer's payoffs are determined by differences between the underlying asset price at expiration and the strike price.
- **Options Type** – there are two common types of options, put and call.
 - a) the put option grants the options buyer the right to sell the underlying asset at the strike price.
 - b) the call option grants the options buyer the right to buy the underlying asset at the strike price.

- **Options Premium** – price the options buyer pays the options seller in exchange for the right to exercise an option. Since the options contract is a traded asset like stocks, it changes depending on market conditions and price of the underlying asset.

1.4.1 Strangle

The strangle strategy consists of selling both a call and a put with the same expiration date, same underlying asset, but different strike prices. Strangle strike prices are usually chosen “out of the money,” meaning the call strike price is greater than the market price at trade date, while the put strike price is less than the market price at trade date. Strangle payoffs are the sum of the call and put option payoffs (Smith). The options writer receives maximum payoff when the asset price at expiration is in between the call and put strike prices.

1.4.2 Payoffs

Payoffs are determined by the difference between the asset price and fixed strike price at the expiration date. If the options buyer holds a put option with a strike price higher than its asset price, the buyer will purchase at the current market price then sell at the strike price, earning the difference. The options writer loses what the options buyer gains. On the other hand, if the options buyer holds a call option with a strike price lower than the asset price, he buys the asset at the strike price then sells to the options writer at the higher market price. The options buyer only chooses to exercise a contract if he stands to profit from it. The options seller earns the premium but can lose money if an option is exercised (Hull).

The put and call option profit from the short position are:

$$\begin{aligned}
 \text{put profit} &= \begin{cases} p_0 + s_t - k_p, & k_p > s_t \\ p_0, & k_p \leq s_t \end{cases} \\
 \text{call profit} &= \begin{cases} c_0 + k_c - s_t, & s_t > k_c \\ c_0, & s_t \leq k_c \end{cases} \quad \text{Where:}
 \end{aligned}$$

k_p : put strike price

k_c : call strike price

s_t : stock price at maturity

p_0 : put option premium at contract writing

c_0 : call option premium at contract writing

Figure 1 is an example that shows the potential profit the options buyer gains from the option at expiration when the option is exercised. The blue line is the S&P 500 index price from 11/17/2009 to 12/18/2009 (December expiration date). The red horizontal line represents the call strike price. If the index price closes above that line then the buyer exercises the call option. Similarly, the green line represents the put strike price. If the index closes below the line, the buyer exercises the put option.

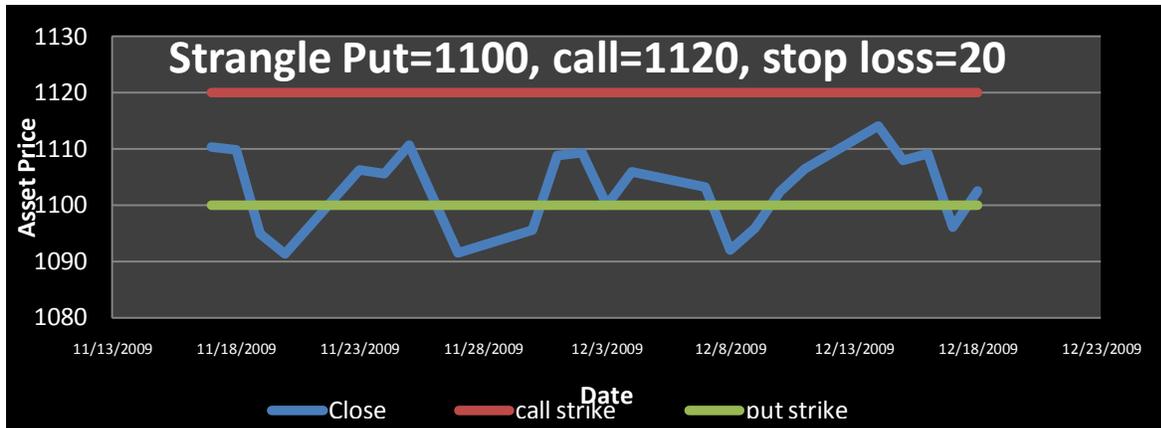


Figure 1

The corresponding payoff graph is shown in Figure 2. The dark curve is the payoff of the two options summed together with the x-axis S&P 500 index price at expiration. The maximum payoff for the seller is received when the asset price is in between the put and call strike prices at expiration. The lighter line shows the strangle payoff with the addition of initial premium received and illustrates the overall payoff for the strategy.

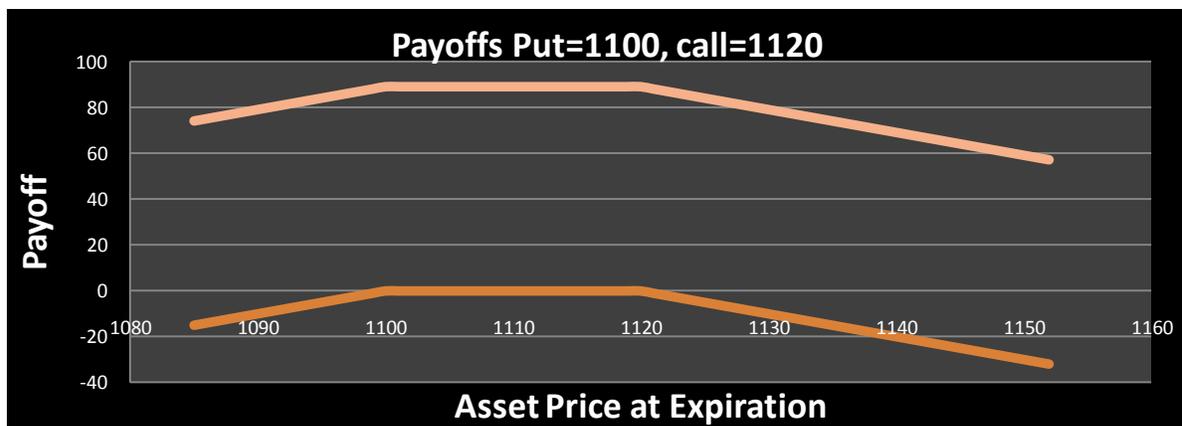


Figure 2

1.4.3 Stop-loss

Stop-loss is an investment policy used to limit losses by options writers (option sellers), which is usually stated as a fixed price difference in premium price. Once the market premium price of an option increases by the stop-loss amount, the options writer buys back the identical option so he is no longer subject to risk of loss (Smith). The loss the options writer takes is the stop-loss amount, assuming the writer can instantaneously buy the option back. A strangle strategy combines the writing of two options so it is possible for the writer to hit two, one, or no stop-losses.

For example, Figure 3 shows a put option with strike price 1100 and a call option with strike price 1120. To illustrate a stop-loss policy of 15, add in a call stop-loss line and put stop-loss lines which are calculated by adding 15 to the options premiums. If at any time an options premium crosses its corresponding stop-loss line, a stop-loss is executed. An identical option is bought and the writer takes a 15 point loss. In this particular example neither stop-loss is hit.

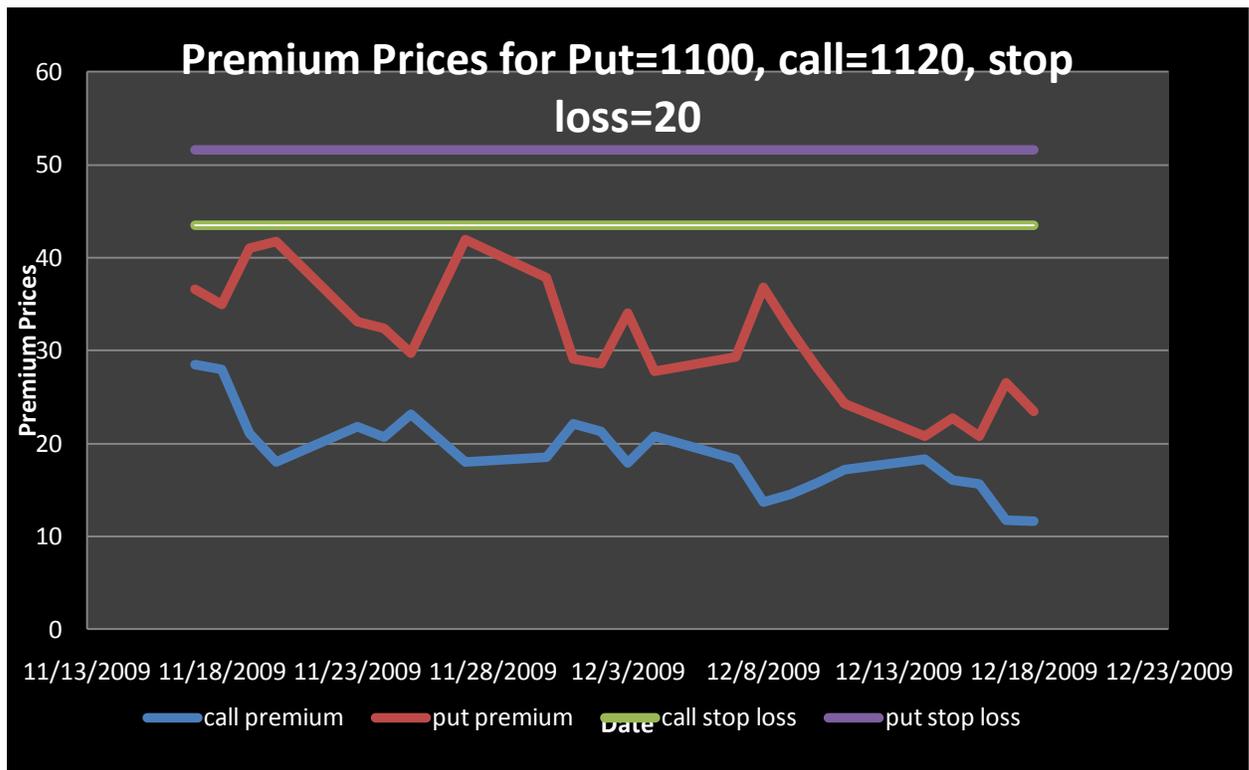


Figure 3

1.4.4 Convergence of Options Premium and Payoffs at Expiration

Since the premium of an option is derived from the potential gain of the underlying asset, the premium will equal the options buyer's payoff at expiration.

$$\text{put premium at maturity: } p_t = \begin{cases} k_p - s_t, & s_t < k_p \\ 0, & s_t \geq k_p \end{cases}$$

$$\text{call premium at maturity: } c_t = \begin{cases} s_t - k_c, & s_t > k_c \\ 0, & s_t \leq k_c \end{cases}$$

1.4.5 Margin Account

When investors trade stock, the maximum loss occurs when the stock price falls to zero and the entire capital is lost. Options writers do not buy any asset therefore their risk is transferred to their broker. The margin account is instituted to hold funds as collateral. Options traders are required to have \$5000 in their margin account for each options contract traded at all times which takes the place of principal. If there is increased risk of loss, the options trader is subject to a margin call. He is required to add more funds to his account. If the trader chooses not to add to the margin account, his broker liquidates the options and closes his position.

1.4.6 Fractional investment

Investors face a trade-off between low profit and high chance of catastrophic loss when determining how much of their principal to invest. Potential profits increase as investors choose to invest because there are fewer assets sitting idle in an account. On the other hand, over-investing assumes more risk because larger losses are sustained when assets lose value. The middle ground is a continuum depending on how much risk the investor is willing to assume. Fractional investment is a method of determining how many trades to make, based on equity and percentage of investment. For example, options require a \$5000 margin account balance for each contract. The trader writes the number of options, so the margin account requirement is equal to the fraction of equity risked. This fraction is defined as the "optimal f" for the investor.

1.4.6.1 Risk of Ruin in Fixed Fractional Trading

Ruin is a subjective state which varies from one trader to another, rendering it difficult to define precisely. For the purposes of this model, ruin is defined as the state where a trader has

lost half of his initial investment. The risk of ruin for a strategy is a value between zero and one that expresses the probability of that strategy losing half of an investment.

In fixed fractional trading, one trades contracts based on a fraction of available money instead of using a constant number of contracts. If a trade generates positive returns, there will be more money available for the next trade resulting in a higher number of contracts. Negative returns, on the other hand, result in fewer contracts for the next trade. If a particular strategy tends to lose money, this behavior has the effect of scaling money invested down over time, making it theoretically impossible to reach zero. For example, if an inventory starts with \$100 then invests and loses half of his money, for the next trade he will have \$50 and invest \$25. (Vince 1990)

Selling contracts requires a certain fixed amount of money. This leads to a practical state of ruin where an investor lacks enough capital to "trade one base unit." (Vince 1990) However, it is unlikely that an investor will follow a losing strategy until this point, which motivates the definition of ruin as losing half of initial capital.

According to Chamness, "markets usually have fat-tailed distributions" and see returns far from the mean more often than most random data sets. This fat tail means ruin is particularly sensitive to the standard deviation of returns. In a proper model for measuring risk, increasing the standard deviation of return values should result in an increase in the risk of ruin. (Chamness 2009)

One can modify the Cox-Miller equation for fixed fractional position sizing. In this equation, the risk of losing one standard deviation is $e^{-\frac{2*a}{d}}$ and the number of times d must be lost to lose an amount z is given by $\frac{z}{d}$. In the following formula, the risk of losing one standard deviation is $e^{-\frac{2*a}{d}}$ and the number of times d must be lost to lose the amount z is given by $\frac{\ln(1-z)}{\ln(1-d)}$. (Chamness 2009)

For fixed fractional position size, the risk of ruin formula is:

$$\text{Risk of ruin: } R = e^{\left[\frac{-2*a}{d} * \frac{\ln(1-z)}{\ln(1-d)}\right]}$$

Where:

z = fraction of lost account, 0.5 for ruin

a = average return

d = standard deviation of return

1.5 Volatility Smile

The implied volatility of an option is the volatility implied by the premium of the option. Implied volatility is variable based on strike price such that “at the money” options are lesser than “out of the money” and “in the money options.” The resulting graph appears to be a smile thus the name. The previous group estimated options premium using a fixed volatility rather than a curve, therefore their premium estimates are incorrect. By using real data instead of calculating premiums using the Black-Scholes model, volatility skew error is avoided. An example of volatility smile curve can be found in the appendix.

1.6 Previous Work

In the Fall 2009 semester, a group of Systems Engineering and Operations Research students at George Mason University worked on a Master’s project to find optimal investment strategies. They used the Black-Scholes model to calculate options premiums of various strike prices for S&P 500 index futures from 1998 to 2008. They then used these premium prices to determine which strategies yield highest returns. We are extending their project by using real data in order to resolve the volatility skew problem and find better results. Furthermore, we are implementing more modifications to the short strangle strategy, finding an optimal fractional investment allocation, and displaying all of our results in a graphic user interface so that decision makers can easily understand the output data.

2 Objectives and Scope

We have three main goals:

- Find short strangle strategies with high average payoff that consistently win
- Find an optimal fraction investment allocation that trades off payoffs and risk of ruin according to investor requirements
- Create a graphical application so users can view and interpret all of our results

2.1 Strangle Strategy Scope

There are numerous possible strategies, each having many positions, variations, modifications, and parameter values. The scope of this project has been restricted to short position strangles with the following parameters and modifications.

2.1.1 Option Parameters

- Trading days before expiration (15 to 60 days)
- Call strike price measured in difference from asset price at trading date (+5 to +50 in increments of 5)
- Put strike price measured in difference from asset price at trading date (-5 to -50 in decrements of 5)

2.1.2 Modifications

- Trade and forget – sell strangle and let it expire
- Stop-loss – choose (5 - 50 in increments of 5) and buy option back if it hits the stop-loss
- Volatility based decision making – check volatility of S&P 500 index to determine whether to trade (maximum volatility index thresholds are set at 30, 40 and 50)

2.1.3 Data Limitations

- Only consider options where the underlying asset is the S&P 500 futures
- Only use options data from 2004 to 2009 because prior years are too sparse to be useful

3 Technical Approach

This project uses an optimization model that mimics the options strategies, trading, and resulting payoffs. The model uses option strike price and index data, along with user inputs for parameters. It outputs all strategies with payoff data and indicators of how strategies perform over time. The optimization portion of the model searches the entire feasible set of strategies, calculates returns, and exports results into binary files. These can be loaded by a graphical user interface to sort and filter the strategies based on criteria, such as risk or return. The optimal strategy is the scenario where returns are high, risk of ruin is low, and the strategy is robust such that small changes in parameters will not greatly change the resulting payoff.

3.1 Assumptions

American options only – This project only considers American options. Since American options offer more flexibility than European options to the options buyer, the premium for American options is greater. Each option is simulated as if it is not exercised until expiration, like a European option. The premium data is for American options, so the simulated payoffs are only valid for American options.

Use of calendar days instead of trading days – There are many days missing from the strike price data due to weekends and holidays. We determined that bias could be introduced if we used trading days, since it is likely that market tendencies shift for different days of the week. For example, 18 calendar days before expiration is always a Monday and if there is no holiday the same day is 14 trading days before expiration. However if there is a holiday, then 14 trading days before expiration becomes the preceding Friday or 21 calendar days before expiration. If we were to use trading days, we suspect that having Fridays mixed in with Mondays for 14 trading days before expiration would skew our results. Therefore, we use calendar days rather than trading days and in the event of a weekend or holiday, the model skips to the next trading day.

Only consider end of day prices – In reality, options writers can trade options continuously throughout the day as long as there is another trader willing to purchase at the premium. Only end of trading day data is available so the model only makes end-of-day trades. A result of this assumption is the model yields less optimal results because the feasible region is greatly reduced by omitting all midday prices. The number of stop-losses is understated since the options writer

may hit a stop-loss in reality, but the model does not execute the stop-loss due to low precision data.

Do not consider nominal interest rate or inflation – Real interest rates are not simulated because they vary over time and accurate inflation rate data are not readily available. Inflation rates are commonly quoted by price indices, but these values are only estimates and not the actual inflation rate. The resulting payoffs will be slightly inaccurate, as assets and currencies devalue over time. We model an options writer selling an option at a specific time and then calculate payoffs at a later time. Values from both times are used to determine profit but the result does not account for time value of money. Depending on the real interest rate (nominal interest rate – inflation rate), profits may be overstated or understated.

Do not simulate slippage – Slippage is the difference between estimated trade price and actual trade prices. It occurs when a trader wants to make a trade at a certain price but ends up trading at a slightly different price. Slippage can be caused by broker inefficiencies such as not executing the order quickly or being able to find a willing trading partner. There is no accurate data available to integrate into the model. Therefore we assume there is always a willing trading partner and trades are made at asking price.

Use S&P 500 index prices rather than S&P 500 futures as the underlying asset – The available data are for options with S&P 500 futures as the underlying asset, but only S&P 500 index data is available. Index data is used instead of futures data for calculating strike price differences such as +5 or -10.

3.2 Data

We have E-mini options data from 1997 to 2009 from the Chicago Mercantile Exchange Group to use as input in the model. The data contains information on options premium, options type (call or put), strike price, expiration date, and volatility. Twelve years of data was provided, however only a few years were meaningful since 1997-2003 are sparse. Missing data is not interpolated so the scope is limited to 2004-2009 with the focus on 2007-2009.

In order to index strike prices by difference from asset price, the end of day prices for the S&P 500 Futures was needed. But due to unavailability, the S&P 500 index from Yahoo! Finance is substituted. We were able to recover S&P 500 Futures prices on expiration dates using the premium data and Figure 4 shows how well S&P 500 index prices acts as an approximate.

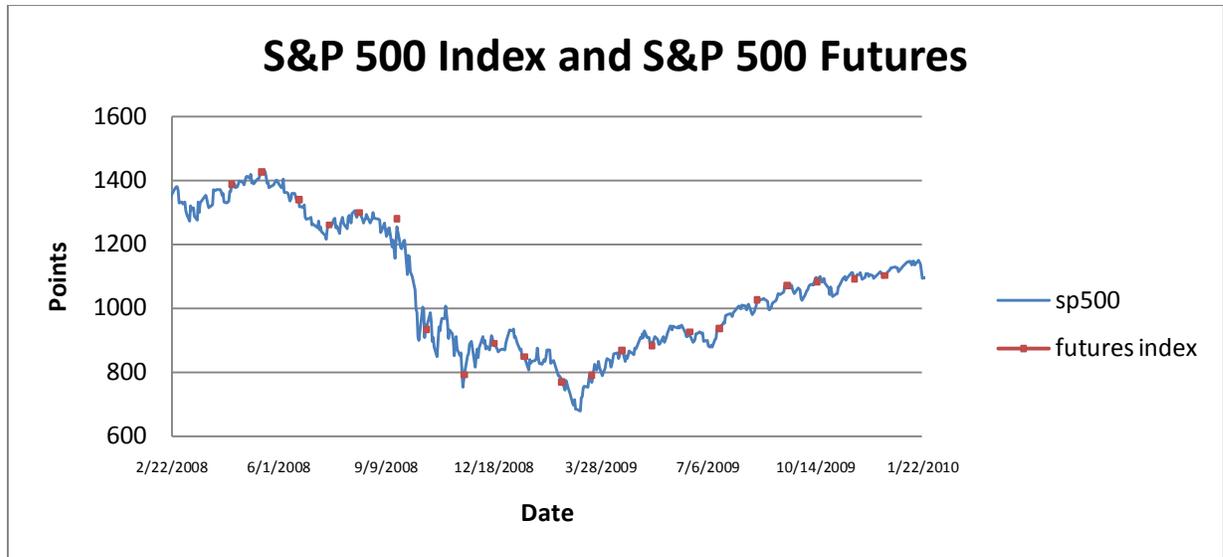


Figure 4

Using volatility driven decision making requires S&P500 volatility data, so that the optimization model can determine when volatility is at an acceptable level. We were also able to find volatility index data from Yahoo! Finance.

4 Model and Architecture

4.1 Preparing Data for Model

The data came in large CSV flat files that contain all the data fields as well as, data that was not useful for our model. Before beginning the modeling process we first arranged the data into a more intuitive structure. An options object was created first which parses the data in each row and stores the following information:

- Trading date
- Option type, call or put
- Strike Price (strike price is described as difference from asset price at trading date. To calculate this difference the asset price is first rounded to the nearest integer, then the next greater multiple of 5 becomes the +5 call and the next lesser multiple of 5 becomes the -5 put)
- Expiration date (the raw data file contains only expiration month and expiration year. Since we know that expiration dates fall on the 3rd Friday of each month we can calculate the expiration date)
- Premium
- Days before expiration (this value was calculated by finding the difference between the trading date and expiration date)
- Then the options objects are arranged in a hash map with expiration date as key and options objects as values. Since our model loops by expiration date on the highest level the hash map simplifies searching for the correct data.

4.2 Modeling Strategies

Trade and Forget - Trade and forget is the most basic version of strangle that we model. The options writer sells an option and waits until expiration. This is the basis for more advanced strangle strategies. The trade and forget simulates selling strangle for each month and calculating the payoff. The algorithm for all three models can be found in the appendix.

Strangle with Stop-Loss - Strangle with stop-loss adds one layer of complexity to the model. We execute the stop-loss as soon as the market premium price increases by at least the stop-loss value, at which point we buy back the option. To implement stop-loss into the model

we need to check the premium price for all trading days between the options writing date and expiration date.

Volatility Driven Decision Making - Volatility driven decision making prevents investors from investing when the market is too volatile. Payoffs can become unpredictable in a volatile market so risk averse investors might not trade until the volatility drops to acceptable levels.

4.3 Modeling the Optimal Fractional Investment Allocation

The model sets fractional allocation values, in multiples of 5% from 5% to 100%, and calculates payoffs for the entire period of input data. By limiting fractional allocation values, precision is lost but model run time is gained. Missing values can be estimated since the curve of the function is smooth.

After initializing the trading account as a fraction of capital, the next target is to identify how many options to write or sell. Here the consideration is the largest loss and the margin requirement, which is \$5000 for selling an option from the commission. The function can be expressed as follows:

$$\text{The Number of Contract} = \frac{\text{The trading account}}{\max(\text{the largest loss}, \text{margin})}$$

We compute final payoffs for all fractions for each strategy and identify the fraction with the highest payoff as optimal.

4.4 Graphical User Interface

A major component of this project consists of a decision support system for a hypothetical option writer in the form of a graphical user interface (GUI). This interface allows a decision maker to browse returns for all strategies over a selected trading period, along with associated data such as average percent return, optimal fraction of investment, and worst draw-down. The decision maker can filter strategies based on days before expiration, put and call prices, stop-loss value, and maximum volatility. It allows the decision maker to manage risk by filtering strategies that exceed a maximum worst draw-down during the examined trading period, exceed a maximum risk of ruin, or cause the seller's account to dip below a given value.

When the model runs a set of strategies over a given time period, it computes individual trade returns for each strategy along with associated metrics. The output of each trade is written to memory as a data record containing just these values. These data are then saved to a file as a serialized Java object for use later. It is possible to generate and store multiple files with strategies of different types taking place over differing periods of time.

Just as the model and data parsing components, the GUI itself is written in the Java programming language, enabling it to load files written out by the model. Upon start, it loads a user-specified output file from the model containing the trade return data into memory. It then loads the available strategy parameters for days before expiration, put price, call price, stop-loss value, and maximum volatility from the data file into drop-down filters for the user to find strategies. When clicking the filter button, matching strategy data are loaded into a table in the GUI, and allows the user to sort by the available metrics. The user can also select data and copy it into a document or spreadsheet.

The screenshot shows a Java Swing window titled "S&P 500 Investment Allocation". On the left side, there are several filter controls with "From:" and "To:" labels and dropdown menus:

- Days Range:** From: 29, To: 31
- Put Range:** From: -25, To: -5
- Call Range:** From: 5, To: 25
- Stop-Loss Range:** From: 5.0, To: None
- Max Volatility:** From: None, To: None
- Min Worst TWR:** (empty text box)
- Max Draw-Down:** (empty text box)
- Max Risk of Ruin:** (empty text box)

At the bottom left is a "Filter" button. Below the table, it says "Filter returned 750 strategies". At the bottom right is a "Plot Strategy Returns" button.

Days	Put	Call	Stop-L...	Optim...	Final TWR	Worst TWR	Draw...	Risk of...	Avg % Ret...	Av...	% Re...	% Wins	Avg ...	Avg L...	StdD...	Shar...
31	-15	5	20	100	95.755	1.014	18.8	0	11.2	14.414	24.8	83.3	18.054	-3.785	12.7...	1.132
31	-5	5	20	100	73.435	1.127	35.8	0.4	10.2	13.903	22.5	80.6	19.252	-8.259	11.79	1.179
31	-15	10	20	100	81.006	1.019	21.1	0	10.8	13.85	25	86.1	16.9	-5.06	12.5...	1.108
31	-20	5	20	100	76.723	0.971	20.7	0	10.7	13.731	24.3	80.6	17.997	-3.944	12.5...	1.093
31	-10	5	20	100	64.475	1.066	35.8	0.5	9.8	13.561	22.6	80.6	18.825	-8.244	12.6...	1.069
31	-5	10	20	100	62.699	1.132	33.6	0.3	9.9	13.339	22.6	83.3	17.946	-9.7	11.7...	1.132
31	-15	15	20	100	67.4	1.003	23.3	0	10.4	13.267	25.2	86.1	16.381	-6.04	12.2...	1.082
31	-20	10	20	100	65.104	0.977	23	0	10.3	13.167	24.5	83.3	16.807	-5.033	12.2...	1.075
31	-25	5	20	100	62.376	0.933	22.4	0	10.1	13.106	23.9	80.6	17.401	-4.687	12.7...	1.032
31	-10	10	20	100	55.243	1.072	33.6	0.4	9.5	12.998	22.8	83.3	17.534	-9.683	12.5	1.04
31	-5	15	20	100	52.602	1.113	31.3	0.2	9.5	12.756	22.6	83.3	17.333	-10.1...	11.7...	1.084
31	-20	15	20	100	54.228	0.96	25.2	0	9.8	12.584	24.7	83.3	16.271	-5.85	11.9...	1.054
31	-25	10	20	100	52.95	0.939	24.7	0	9.8	12.542	24.1	83.3	16.23	-5.9	12.2...	1.021
31	-15	20	20	100	51.924	0.977	25.3	0	9.7	12.503	24.9	80.6	16.742	-5.057	12.2...	1.024
31	-5	25	15	100	41.505	1.187	30	2.8	8.4	12.443	24.2	77.8	19.362	-11.7...	12.4...	1.002
31	-5	15	15	100	37.487	1.235	51	8	7.8	12.436	22.1	77.8	20.357	-15.2...	12.4...	1.002
31	-10	15	20	100	46.542	1.054	31.3	0.3	9.1	12.414	22.8	83.3	16.921	-10.1...	12.3...	1.007
31	-5	25	25	100	39.249	0.902	34.8	0.8	8.5	12.318	23.9	69.4	21.238	-7.955	15.47	0.796
29	-5	20	40	100	23.493	0.673	47.8	20	6.2	12.218	24.2	62.9	28.169	-14.7...	17.9...	0.679
31	-5	10	15	100	34.065	1.246	51	8.8	7.6	12.115	20.5	77.8	19.93	-15.2...	12.1...	1
31	-5	15	10	100	38.071	1.235	36	2.2	8.3	12.09	21.5	83.3	18.508	-20	11.1...	1.087
29	-5	25	40	100	21.562	0.711	46	25.6	5.9	12.083	25	62.9	28.45	-15.6...	17.83	0.678
31	-5	5	25	100	43.358	0.959	26.8	0	9.1	12.072	19.5	75	18.256	-6.479	12.2...	0.983
31	-5	5	10	100	35.136	1.227	48.8	4.3	7.9	12.062	19.5	80.6	19.801	-20	11.8...	1.015
31	-5	20	20	100	40.923	1.085	28.8	0.3	8.8	11.992	22.3	77.8	17.686	-7.938	11.9...	1.002
31	-25	15	20	100	44.229	0.924	26.8	0	9.3	11.959	24.2	83.3	15.694	-6.717	11.91	1.004
31	-5	20	25	100	37.78	0.919	33.1	0.2	8.6	11.886	22.1	69.4	19.96	-6.464	13.6...	0.872
31	-10	25	15	100	34.544	1.173	30	3.2	8	11.846	23.9	80.6	18.023	-13.7...	12.47	0.95
31	-10	15	15	100	31.11	1.221	51	9	7.4	11.839	21.7	80.6	18.983	-17.7...	12.6...	0.937
31	-20	20	20	100	41.827	0.936	27.1	0	9.2	11.82	24.4	77.8	16.636	-5.038	11.8...	0.997
31	-5	20	15	100	30.841	1.209	50.9	8.7	7.4	11.817	21.9	75	20.304	-13.6...	12.6...	0.937
31	-25	5	5	100	36.141	1.092	27.1	0.6	8.4	11.799	21.5	69.4	21.39	-10	11.9...	0.985
31	-5	5	30	100	34.209	0.808	38.4	0.6	8.2	11.761	19	77.8	17.921	-9.801	15.3...	0.768
31	-15	25	20	100	39.618	0.962	27.1	0	9	11.727	24.5	72.2	18.026	-4.65	12.0...	0.976
31	-5	15	30	100	30.877	0.793	42.2	2	7.7	11.722	20.8	72.2	20.115	-10.1	16.1...	0.724

Figure 5

From Figure 5 the user can select multiple strategies and compare them by the available statistics. In order to view returns of one or more selected strategies, the user can click on the "Plot Strategy Returns" button. That opens the strangle returns view in Figure 6:

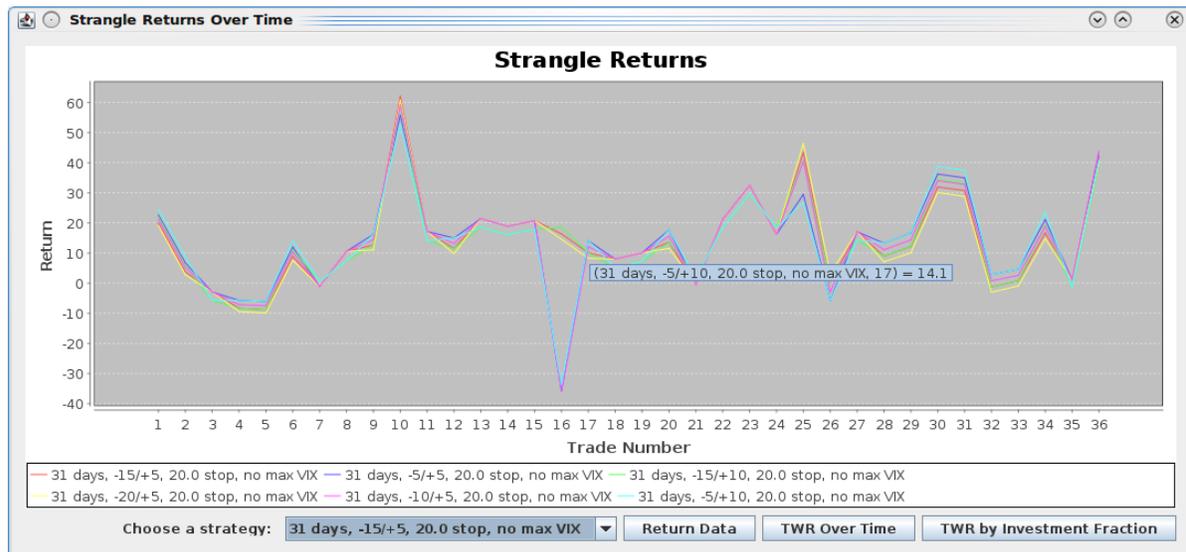


Figure 6

Figure 6 shows actual modeled returns by trade for each selected strategy. The horizontal axis shows trading opportunities as they occur over time and the vertical axis represents the returns. It allows the user to select three different views of the data: the actual return data, terminal wealth relative (TWR) over the trading period, or the final TWR after trading based on fraction of investment.

Trade Number	Expiration Date	Payoff
1	2007-1-19	20.1
2	2007-2-16	3.9
3	2007-3-16	-2.9
4	2007-4-20	-8.3
5	2007-5-18	-8.8
6	2007-6-15	8.91
7	2007-7-20	-1.1
8	2007-8-17	10.6
9	2007-9-21	12.7
10	2007-10-19	62.1
11	2007-11-16	17.2
12	2007-12-21	11.5
13	2008-1-18	21.5
14	2008-2-15	18.9
15	2008-3-21	20.8
16	2008-4-18	16.4
17	2008-5-16	10.1
18	2008-6-20	8
19	2008-7-18	10.1
20	2008-8-15	13.6
21	2008-9-19	-0.41
22	2008-10-17	21.2
23	2008-11-21	32.5
24	2008-12-19	16.26

Figure 7

Figure 7 shows the actual return data that is displayed for a strategy in the strangle returns window. Each trade has an expiration date associated with it and a payoff for selling it. The payoff accounts for both put and call premiums, as well as any stop-loss that is triggered. If a trade cannot be performed, i.e. a holiday, that trade is left out. The payoff values are in points.

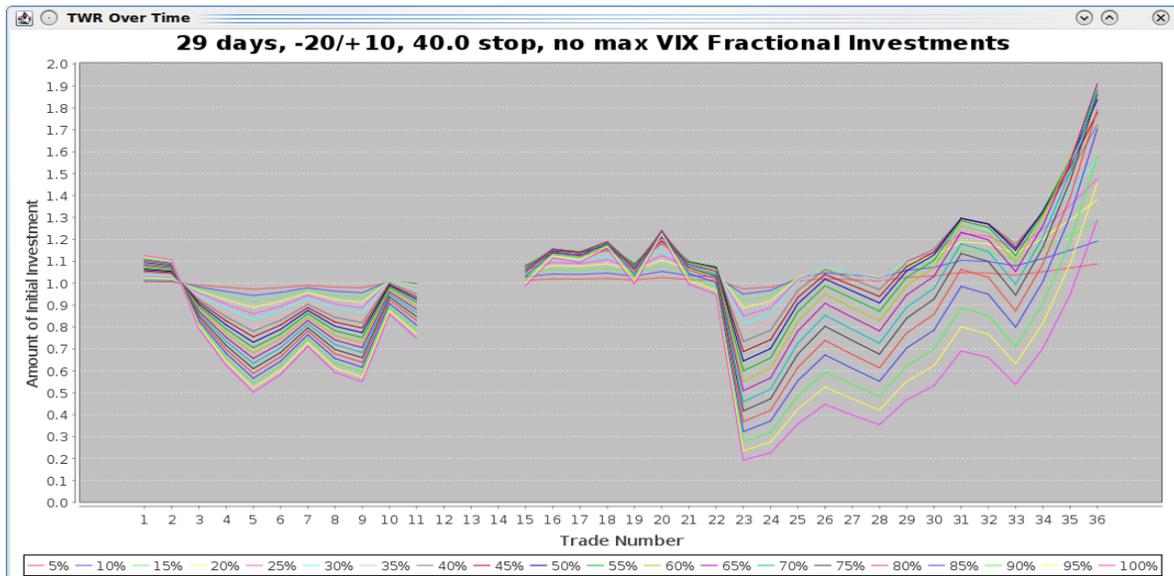


Figure 8

Figure 8 shows the TWR of a particular strategy over time. The strategy always starts with a TWR of one, representing the initial investment. The horizontal axis represents trades and the vertical axis is TWR. Each line represents a different fraction of investment. For instance, the first line indexed on the bottom left in the legend, refers to the plot of a 5% investment. In the

case of this strategy, investing a higher percentage of one's capital gives a lower final return, due to the large loss from trade 23.

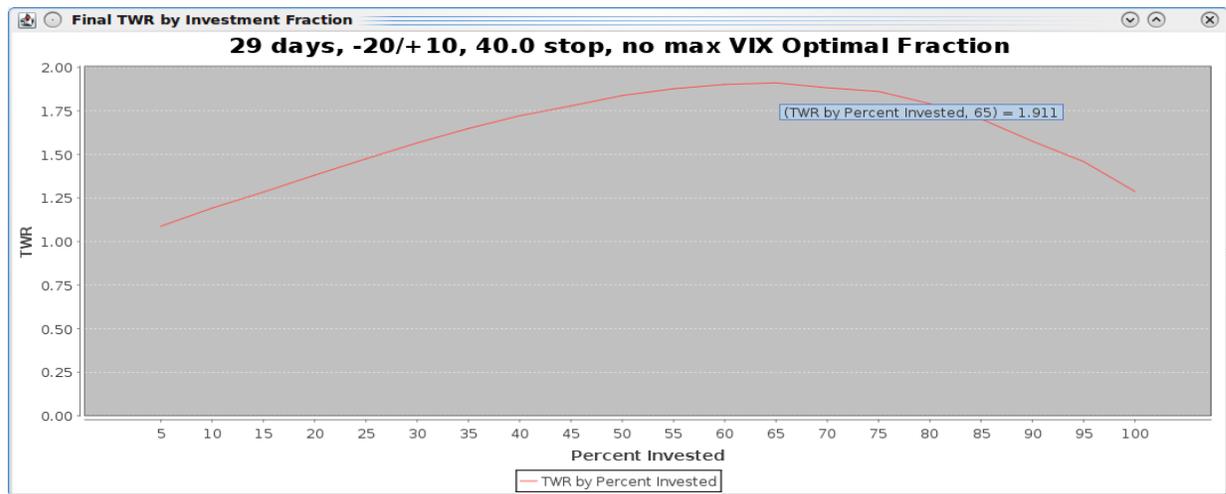


Figure 9

Figure 9 shows the final TWR of based on investment fraction of a given strategy. These are the same data as the final trade of the previous window, and therefore represent the amount of one's initial capital after the entire trading period.

5 Results

The graphics in this section are used to represent strangle strategy performance using two metrics, TWR and percent return. TWR measures the performance of a strategy over multiple trading periods, while percent return is the percent difference in assets over a single trading period. In this section TWR will be measured in multiple of initial investment so $TWR = 5$ means the final equity is 5 times the initial equity. Maximum draw down is the greatest percentage drop in capital between any two points in a particular period. Investors may not necessarily invest at the beginning of an analyzed period so maximum draw down helps to assess risk by showing the maximum possible loss.

Early analysis of the available data showed the most available data are from years 2007 to 2009, with 2004 to 2006 usable but not conclusive, and all prior data unusable. Therefore we perform our analysis first with the overall data from 2007 to 2009 and then, with 2004 to 2006 separately.

5.1 Days before Expiration

First we searched for the optimal trading day before expiration to write options, and then we found it to be between 39 and 44 days before expiration (note: there is no data for 40 and 41 days before expiration because there is no trading on Saturday and Sunday). Figure 10 shows averages of all optimization model outputs sorted by days before expiration. Each bar in this chart is the average of all strategies (including all variations of parameters and modifications) from 2007 to 2007 with a specific day before expiration.

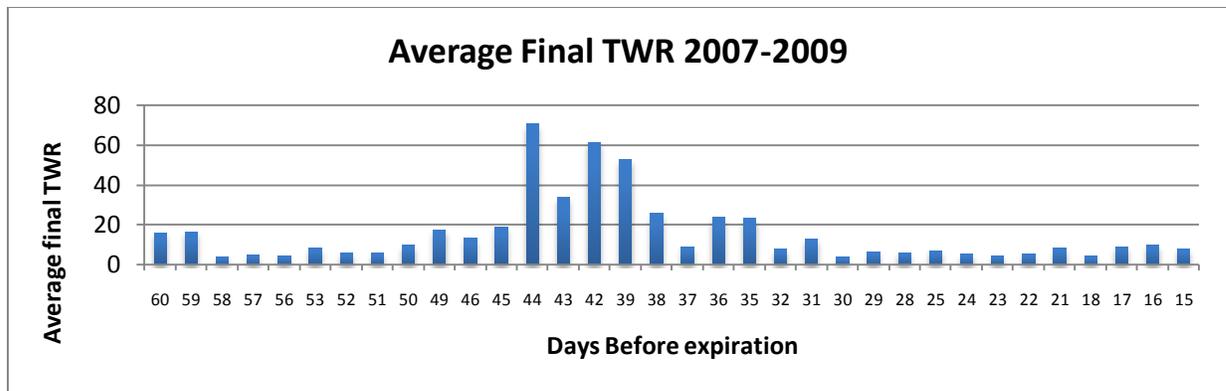


Figure 10

Based on this and many following charts we are confident that (39-44) days before expiration is the best time to sell a strangle strategy for 2007-2009. According to the charts, we begin to narrow down the strategies shown and 39-44 days before expiration are still clearly optimal. Therefore we believe this parameter is not highly dependent on other parameters. We also searched for an optimal selling date for 2004-2006 but there was no obvious optimal date. Figure 11 is for 2004-2006.

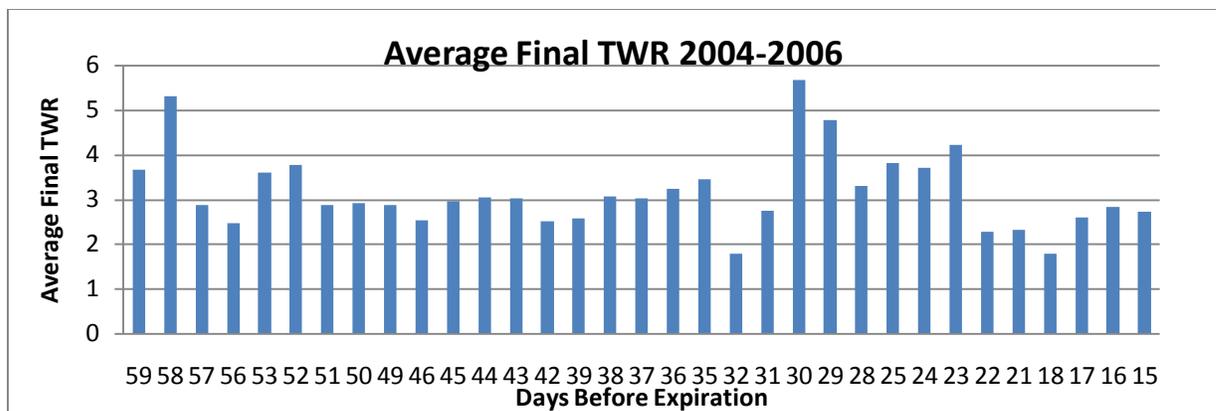


Figure 11

Final TWR is an order of magnitude less than 2007-2008 due to fewer available trades stemming from incomplete data. To overcome the data availability bias we use average percent monthly return to compare the time periods. We compare average monthly returns in Figure 12.

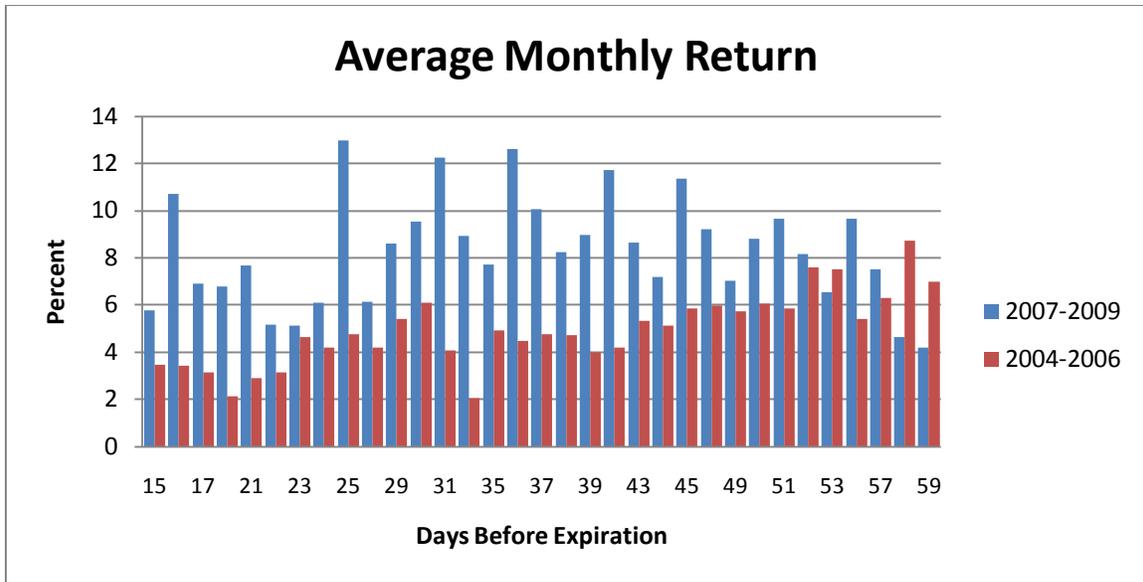


Figure 12

2007-2009 averages dominate 2004-2006 except for three days near 60 days before expiration. In some cases, 2007-2009 averages are at least twice as much as 2004-2006 averages leading us to examine premium prices.

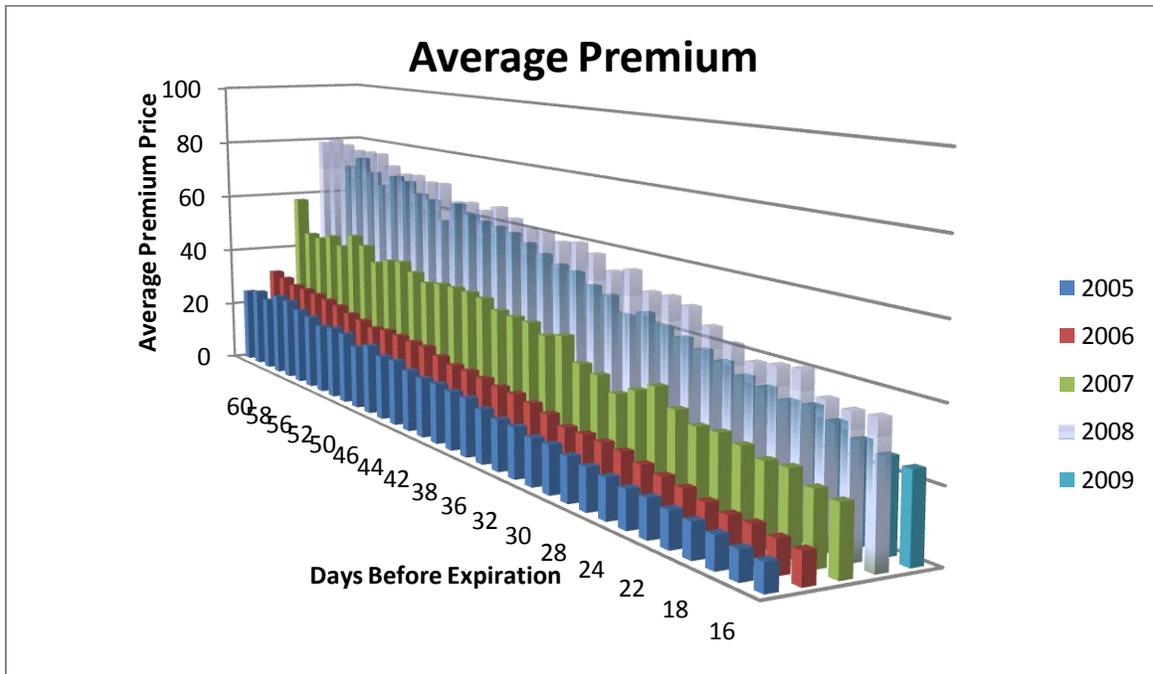


Figure 12

Initial premium received by the options writer is a large factor in final payoffs and Figure 13 shows that 2007-2009 premiums are substantially higher than 2005 and 2006.

5.2 Stop-loss

The next parameter to analyze is stop-loss. Figure 14 separates model outputs by both days before expiration and stop-loss for the period 2007-2009. Each bar in the following chart is the average of all model outputs with fixed day before expiration and stop-loss. The best stop-losses are either 15 or 20. In the optimal trading day range of 39-44 days, stop-loss value of 15 is superior for 39 and 44 days before expiration, while stop-loss value of 20 is superior for 42 and 43 days before expiration.

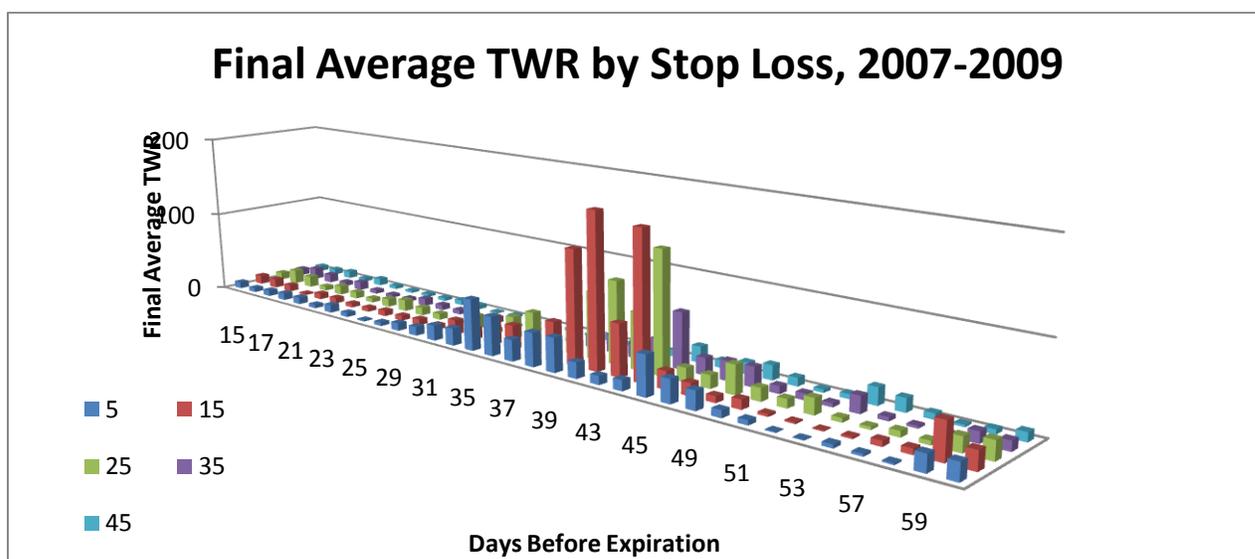


Figure 13

Figure 15 shows a stop-loss comparison for 2004-2007. Note the TWRs for 2004-2006 are much lower than 2007-2009. That is due to missing data in the earlier years. Since the model skips trading for months where data is missing, profitable strategies with fewer trades will result in lesser TWR. Therefore 2007-2009 and 2004-2006 are not comparable as they have different amounts of available data, however stop-loss values in 2004-2006 are comparable to each other.

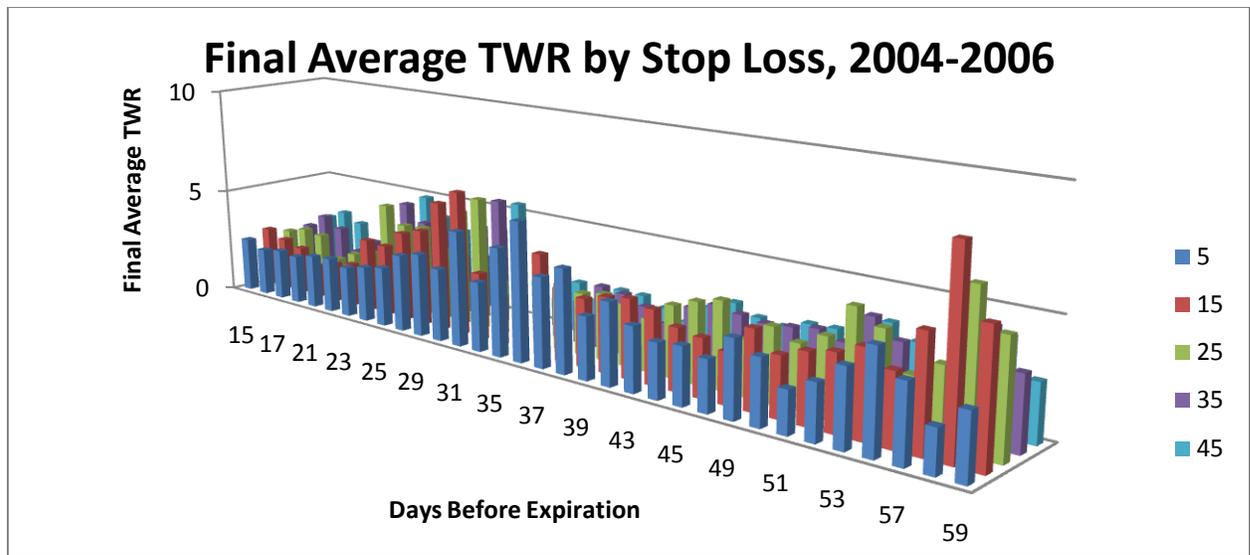


Figure 15

5.3 Strike Prices

We analyzed call and put strike prices individually with the same method used for stop-loss but there are no clear results. Some strike prices are better than other strike prices, so we decided to analyze both strike prices together. We expect that call and put pairs have unique properties that require analysis of both strike prices at once.

The next four surface graphs vary call and put strike prices on the x and y axis, while showing TWR on the z axis. They use data from 2007-2009 with stop-loss fixed at 20. Each of the charts corresponds to a single trading day between 39 to 44 days before expiration.

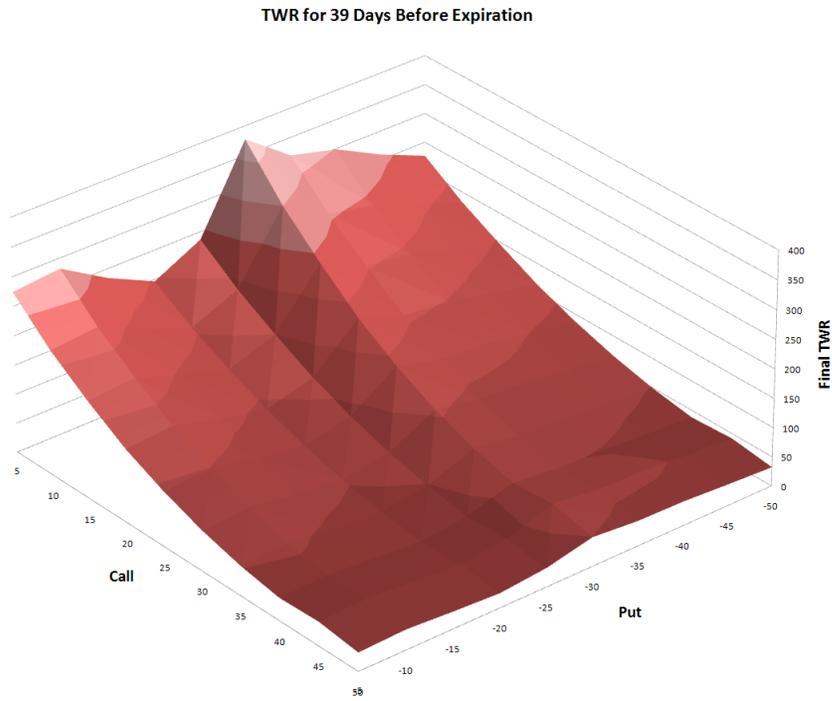


Figure 16

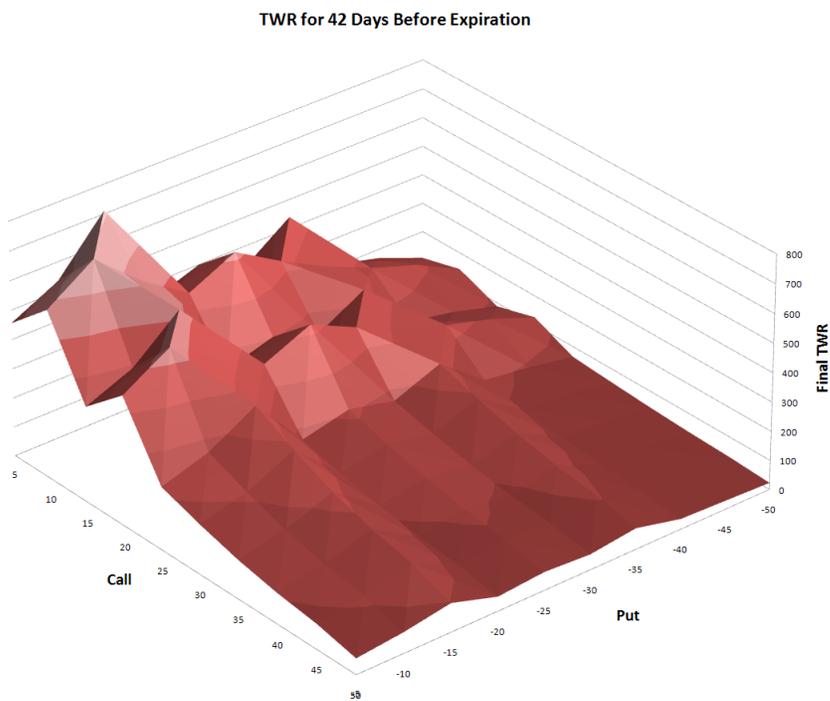


Figure 17

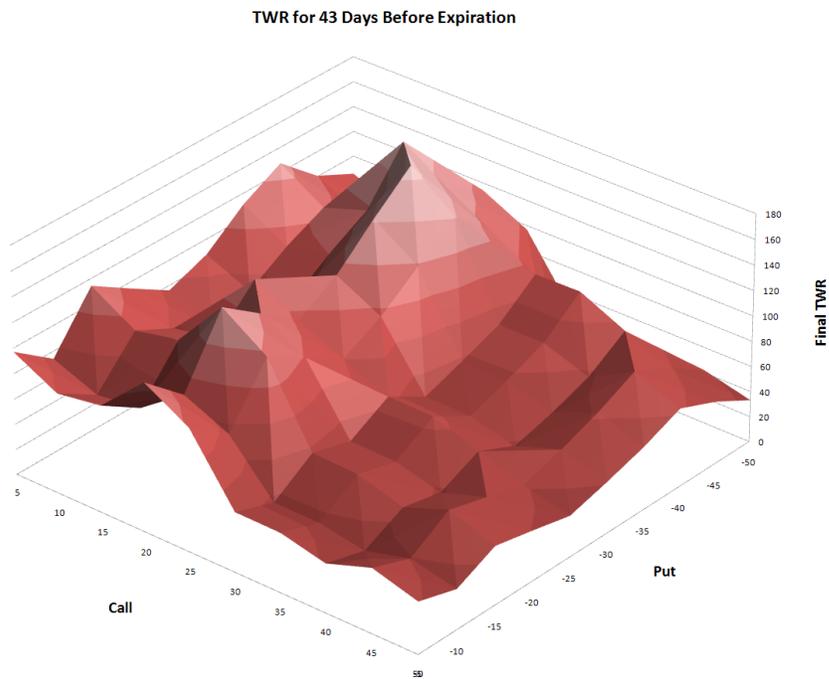


Figure 18

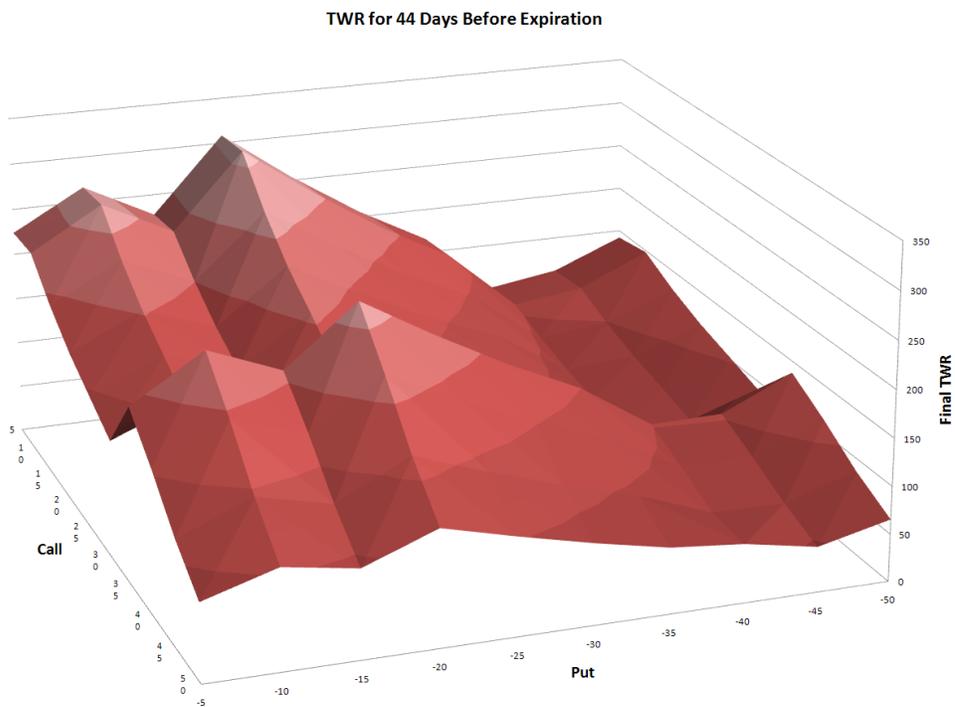


Figure 19

On average, strangles sold 42 days before expiration have the best performance and strangles sold 43 days before expiration have the worst performance. The top three days to sell strangles all have hot spots around call strike +5 to +10 and put strike around -15 to -20. 38 and 44 days before expiration also have hot spots at the same call strike, with lower put strikes around -30. Strangles sold 43 days before expiration on average, have much lower payoffs than the other three trading dates, and also has significantly different hot spot with call strike around +15 and put strike around -40.

5.4 Volatility Driven Decision Making

One modification was to stop trading when S&P 500 volatility is greater than a maximum acceptable value. The following histogram summarizes strategy performance by TWR for maximum volatility of 30, maximum volatility of 50, and no maximum volatility.

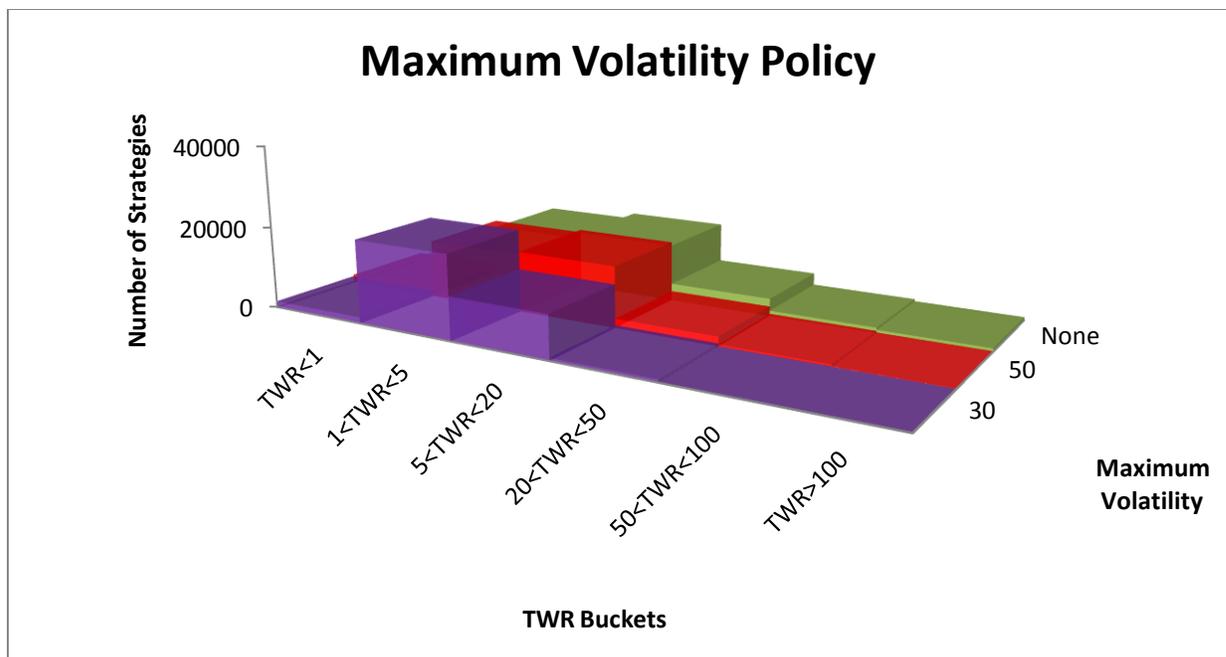


Figure 20

Figure 20 clearly shows that always trading results in better final TWR than any limit on maximum volatility. The performance decreases as the maximum volatility policy becomes stricter. Using maximum volatility of 50 results in lost trades and lost profits but there are still

more than 50% of trades available for all strategies. Using maximum volatility of 30 causes 680 strategies to become sparse, less than 50% of trades are available over the three year period.

5.5 Optimal Strategy

The optimal strategy is for January 2007 through December 2009:

Call strike = +5

Put strike = -15

Stop – loss = 20

Days before expiration = 42

Fraction allocation = 100%

With model outputs:

Final TWR = 711.3

Risk of Ruin = 0%

Average monthly return = 16.4%

Winning trades = 88.2%

Maximum draw down = 15.3

This strategy results in 71130% return on initial investment with no risk of losing 50% of principal. On average, this strategy will return 16.4% per month while winning almost 90% of monthly trades. This result is very profitable but in the evaluations section, we will explain why model results are more optimistic than what real performance may have been.

5.5.1 Sensitivity Analysis

Days before Expiration

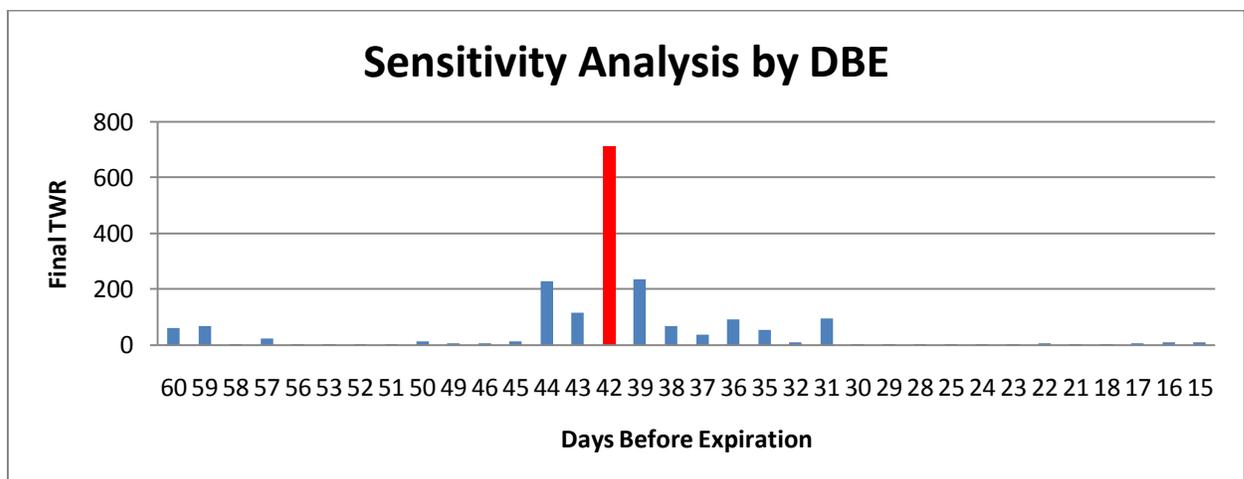


Figure 21

42 days before expiration is the best day to sell this strangle strategy and on average, it's three times better than the second best trade date. However the nearby dates (44 to 39 days

before expiration) are the best of the feasible region. Small changes are expected in sell date to significantly reduce returns, but it is still the best in the feasible region.

Stop-Loss

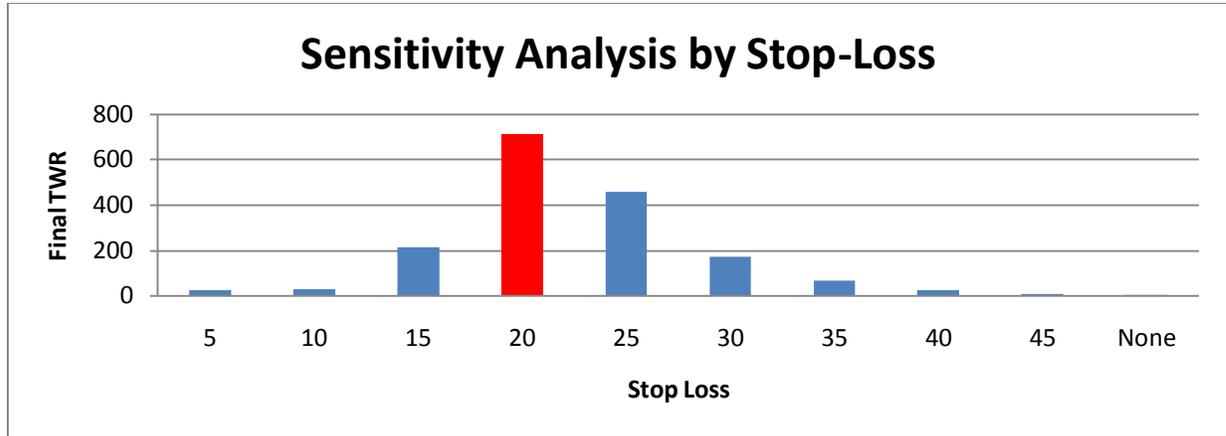


Figure 22

The optimal stop-loss value of 20 is clearly the best, 35% better than the second best stop-loss. The top stop-loss values are near 20, and we can once again expect small changes to significantly reduce returns but still be the best of all available values.

Call Strike Price

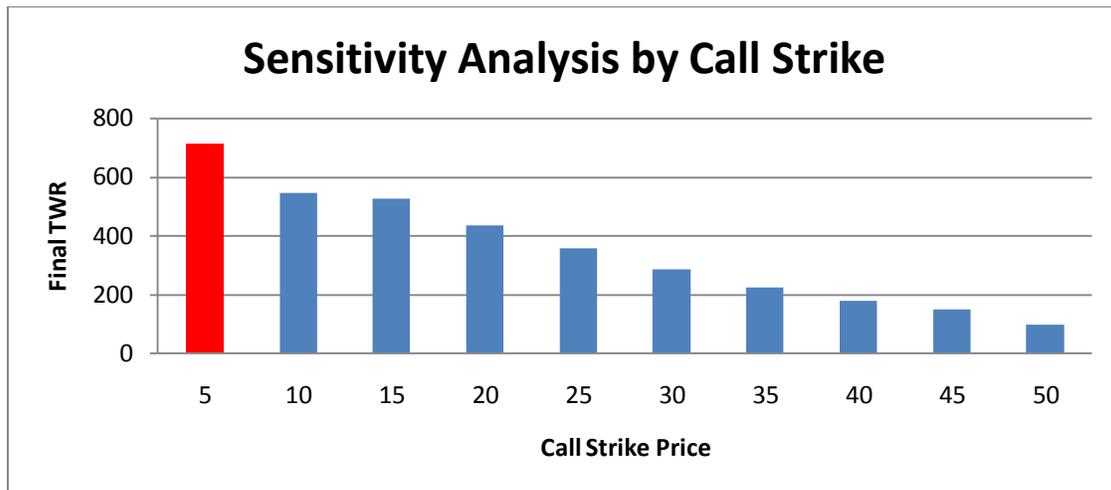


Figure 23

The optimal call strike price of +5 is the best by about 23% with the next best strike prices closest to the optimal. Our analysis shows that the closer to the optimal strike price the better the payoffs, so we can once again expect small changes to still yield good returns.

Put Strike

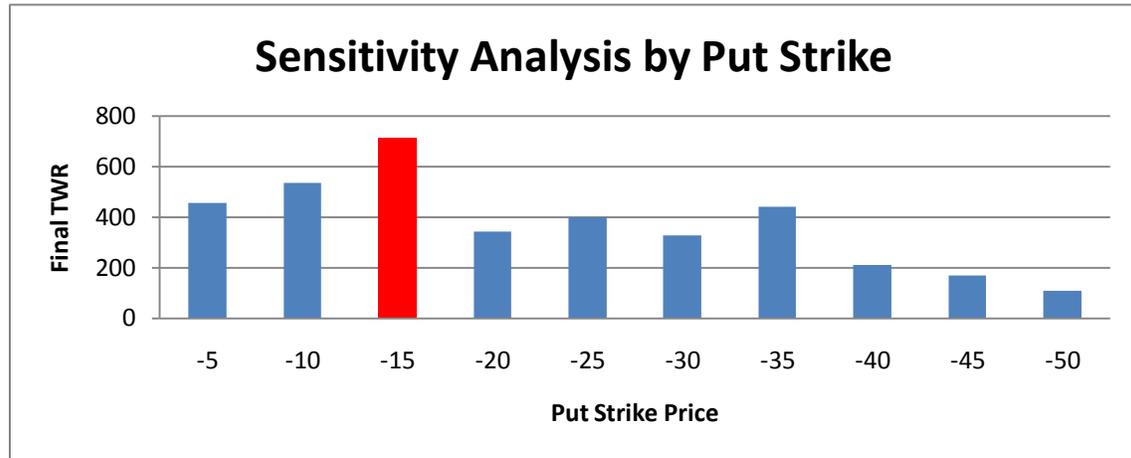


Figure 24

Similar to the optimal call strike price, the put call strike is 25% better than the next best value, however not all of the other top strategies are necessarily close to the optimal. We can expect that some similar put strike prices to be worse than farther put strike price, as -35 performs better than -20 to -30

Sensitivity Analysis Conclusion

In general, the optimal strategy is very sensitive to changes as payoffs can be as much as 65% less. However, we can still expect some of the best strategies available earning at least a 10000% return on investment, if implemented close to the optimal parameters.

5.6 Optimal Fractional Investment Allocation and Risk of Ruin

Figure 25 shows the optimal strangle strategy with different fractional allocations from 2007 to 2009. The x-axis represents trade numbers and since we perform one trade per month each tick represents one month. Each curve represents a fractional allocation percentage. Since trades 20 and 23 are missing, the curves have two gaps.

42 days, -15/+5, 20.0 stop, no max VIX Fractional Investments

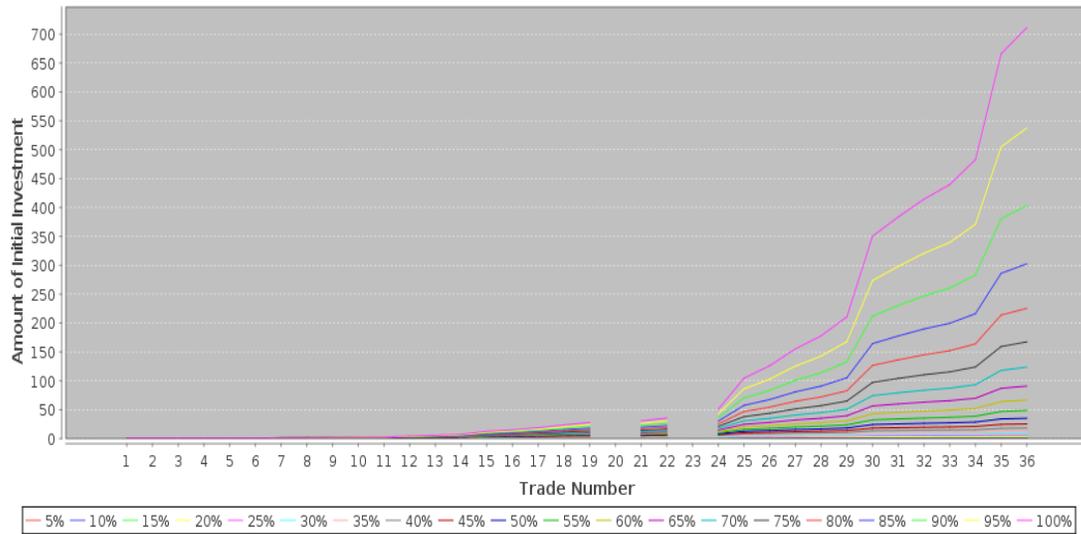


Figure 25

Since this strategy consistently profits each month, the optimal fractional investment allocation is 100%. The risk associated with full allocation of equity is ruin, if there is a large loss in a single month but since this strategy always wins, none of the curves intersect, and higher allocation yields higher TWR. The secondary result of the consistently winning strategy is the associated risk of ruin for all fractional allocations is zero.

6 *Evaluation*

Our analysis yielded convincing optimal choices for parameters for 2007 to 2009. There is clearly a best trade date range to sell a strangle strategy (39-44 days before expiration), a best stop-loss policy to use (15 or 20), and a significant hot spot in the strike price plane (call strike +5 or +10 and put strike -15 or -20). However, these optimal parameters do not match our analysis for the three previous years, 2004-2007. It appears that the best strategies are completely different, therefore our optimal results will unlikely remain optimal for long. In fact, analyzing 2004-2007 data to predict optimal strategies in the later period, would not have even come close to finding the actual optimal strategies. We can therefore conclude that even obvious optimal strategies will only remain good strategies for a short period of time. Our assumption that the past is representative of the future is certainly unrealistic over the long term, with some merit in the short term.

Significantly greater returns in 2007 to 2009 compared to 2004 to 2007 are curious, considering former time period contained one of the worst recessions in history. Analysis of our premium data showed that average premiums were significantly higher in the later period than the earlier period, which is consistent with the rising volatility. High volatility exposes options writers to more risk than low volatility, therefore the market price for premiums rise. We were able to find options that consistently profited month by month in both time periods, but strategies from 2007 to 2009 had greater premiums and therefore profits would be larger. But how could our results show such great potential for profit, when in reality many investors were ruined?

Our model's payoff outputs are likely overstated because we did not simulate slippage. Stop-losses are implemented in our model by checking if the stop-loss is hit and if so, we close the position and assign a loss equal to the stop-loss. In reality, the loss associated with the stop-loss will likely be more. It is unlikely that the options writer can always buy back the options instantly. The overstatement of payoffs is magnified during volatile times because losses due to slippage become greater in a fast moving market. We expect a more realistic result would be using a higher stop-loss than 20 such as 40 which would be easier to manage and monitor.

TWR for top strategies are also overstated because we assumed that the market is scalable. As we progress through the model and our equity increases, we reinvest increasingly more and

thus sell increasingly more options contracts. However, as we sell contracts the premiums will drop as the demand for the particular contracts is met. Our model assumes that the premium we want to sell at is not a function of our trades and that our trade volume is not large enough to affect the market. Our equity increases so rapidly that it is almost certain our trade volume will lower the market price. The lower premiums results in lower payoffs so our final TWR on winning strategies are overstated, particularly towards the later months of our trading period.

Another curious result is our top strategies are fully allocated yet have near zero risk of ruin. Strategies that rarely lose will have optimal fractional allocation and low chance of ruin resulting in the highest TWRs. We are performing a retroactive analysis and it is possible we find strategies that rarely, if ever, lose with near zero risk of ruin. However, few investors are risk-tolerant enough to fully allocate their equity; a more realistic result would use a lower fractional investment allocation.

7 Recommendation

It would be inaccurate to use our exact optimal strategies in the future since it would only remain optimal for a short period of time. A continuously weighted forecasting model with current data should be used to update optimal strategies. Manageable higher stop-loss values should be chosen rather than low stop-loss values which can be difficult to implement in a volatile market. Fractional investment allocation should also be less than 100% because the market is constantly moving and therefore the future is still uncertain. Given the limitations of current optimal strategy, we recommend that future work to refine the search for optimal strategies before acting in the market.

8 Future work

Due to the limited time we had to work on this project, we made simplifying assumptions to make the scope manageable. We also left out many factors that could be relevant. Therefore there are areas for future work to produce better results.

Obtaining a more complete and suitable data set, especially for earlier years, would allow us to find better patterns for forecasting the future. 1997 to 2003 were too sparse to find meaningful

strategies. Even in the later years, we had to remove certain strategies where more than half of the premiums were missing. We managed missing premiums by skipping trades resulting in significantly impacted final TWR. We also used S&P500 index data because we do not have S&P 500 futures data, introducing errors when calculating difference in strike price from the underlying asset.

The model can be improved by adding adaptive logic so that optimal strategies are calculated using only a portion of data. We expect that more recent data is indicative of current market conditions, so a forecasting algorithm can be implemented to assign higher weight to more recent data. The expected result would be a more accurate estimation of current optimal strategies.

Our model manages data by reading and writing binary data and CSV files, so every time inputs were updated or modified, the calculations would generate new files for display. Since we need to constantly reload current model output files, it is inconvenient to ensure that the results are current. Using a database to store output files would help coordinate efforts by multiple people in ensuring that everyone is using the same data.

Currently the model only uses basic short strangle and basic short straddle, but there are more complex strategies that could yield better profits. Future efforts could be put towards implementing Bull Spreads, Bear Spreads, Butterfly Spreads strategies.

Generating outputs take ten minutes or more which hinders work on testing new strategies or modifications. Improving algorithms and a more powerful machine could speed up calculations so that turnaround for results is faster.

Some of our results like zero risk of ruin are unrealistic. We use Chamness's formula which has associated assumptions that can be relaxed for a more accurate risk of ruin. Problems such as ignoring slippage can be accounted for by adding stochastic simulation so that payoffs are not overstated.

One of our assumptions is to ignore inflation and interest rates which introduces error associated with time value of money. Adding in these monetary rates would deflate our results for TWR and percentage return in terms of assets at a single point in time.

9 Bibliography

Chamness, David E. "Minimizing your risk of ruin." Futures (2009).

Hull, John C. Options, Futures and Other Derivatives. Prentice Hall, 2005.

J.L. Kelly, JR. "A new interpretation of information rate." The Bell System Technical Journal (1956): 917-926.

John A. Anderson, Robert W. Faff. "Maximizing futures returns using fixed fraction asset allocation." Applied Financial Economics (2004): 1067-1073.

Rotando, Louis R and Edward O Thorp. "The Kelly Criterion and the Stock Market." The American Mathematical Monthly (1992): 922-931.

Smith, Courtney. Option Strategies: Profit-Making Techniques for Stock, Stock Index, and Commodity Options. Wiley, 2008.

Vince, Ralph. Portfolio Management Formulas: Mathematical Trading Methods for the Furfutures, Options and Stock Markets. John Wiley & Sons, Inc, 1990.

—. The New Money Management: A Framework for Asset Allocation. John Wiley & Sons, Inc, 1995.

Acknowledgements

We would like to thank Dr. Kuo-Chu Chang for his patience and tireless efforts guiding and sponsoring this project, Dr. Kathryn Blackmond Laskey for her meticulousness and thoughtful mentorship throughout the capstone process, and Edwina Chen and Winyan Soo Hoo for their assistance proofreading this document.

10 Appendices

10.1 Appendix 1: Requirements

10.1.1 General Requirements

- 3.2.1.1 The Optimal Option Investment Strategy team shall provide a recommendation on optimal investment strategies and optimal fraction of initial capital.
 - 3.2.1.1.1 The Optimal Option Investment Strategy team shall consider the expected average return and risk of ruin in determining the optimal strategies.
 - 3.2.1.1.2 The Optimal Option Investment Strategy team shall provide a range of optimal strategies based on the level of risk acceptable by the customer.
- 3.2.1.2 The Optimal Option Investment Strategy team shall limit the scope to strangle strategies at the time of trade.
- 3.2.1.3 The Optimal Option Investment Strategy team shall limit the scope to short strangle strategy (sale of put and call options only).
- 3.2.1.4 The Optimal Option Investment Strategy team shall only focus on American options.
- 3.2.1.5 The Optimal Option Investment Strategy team shall be applied to only options with S&P 500 Future Index as the underlying asset.

10.1.2 GUI Model Requirements

10.1.2.1 Input/Output Requirements

10.1.2.1.1 Input Requirements

- 10.1.2.1.1.1 The model shall input premium prices, strike prices, SP500 asset prices, initial capital, stop-loss, day range, maximum volatility, maximum draw-down, minimum worst terminal wealth relative, and maximum risk of ruin in order to provide to recommendations on investments strategies.

10.1.2.1.2 Output Requirements

10.1.2.1.2.1 The model shall output average return on investment, percentage return on investment, risk of ruin, Sharpe Ratio, and optimal fraction of the investment as part of the recommendations on investment strategies.

10.1.2.2 Functional Requirements

10.1.2.2.1 The model shall determine the optimal strategies using highest average return, highest terminal wealth relative, risk of ruin, and level of acceptable risk specified by the user.

10.1.2.2.2 The model shall evaluate the expected return, percentage return, lowest and final terminal wealth relative, Sharpe ratio and optimal fraction of the investments as part of the analyses performed for the client.

10.1.2.2.3 The model shall plot the equity curves of returns, average returns, and terminal wealth relative as a function of time for any user-selected strategy.

10.1.2.2.4 The model shall plot terminal wealth relative as a function of time for different percentages of fractions for any user-selected strategy.

10.1.2.3 User Interface Requirements

10.1.2.3.1 The GUI model shall enable the users to enter their specific inputs.

10.1.2.3.2 The GUI model shall provide a range of days, put and call prices, stop-loss, and maximum volatility for the user to select.

10.1.2.3.3 The GUI model shall filter the strategies based on the user's specific inputs.

10.1.2.3.4 The GUI model shall display the strategies with the associated fields in a spreadsheet.

10.1.2.3.5 The GUI model include an option for the user to sort based on any filed.

10.1.2.3.6 The GUI model shall include an option for the users to plot equity curves as a function of time.

10.1.2.3.7 The GUI model shall enable the user to see underlying data of selected equity curve.

10.1.2.4 Software Requirements

10.1.2.4.1 The model shall use JAVA platform as the back-end programming tool to implement the logic for determining the optimal investment strategies.

10.1.2.4.2 The model shall use JAVA platform as the front-end programming tool to directly interface with the client.

10.1.2.5 System-wide/Technology Requirements

10.1.2.5.1 Reliability

10.1.2.5.1.1 Availability

10.1.2.5.1.1.1 The model shall operate as 24/7 basis upon user's request.

10.1.2.5.2 Performance

10.1.2.5.2.1 The response time of the GUI model to display the optimal strategies based on the user's inputs shall be within 60 seconds

10.1.2.5.2.2 The response time of the GUI model to plot any type of curve for strategies shall be less than 30 seconds per chart.

10.1.2.5.3 Security

10.1.2.5.3.1 The GUI model shall be password encrypted to avoid unauthorized access to underlying data.

10.1.3 Data Requirements

10.1.3.1 The Optimal Option Investment Strategy team shall consider the S&P 500 Index data from 2005 through end of 2009.

10.1.3.2 The Optimal Option Investment Strategy team shall analyze call prices +5 to +50 in increments of 5.

10.1.3.3 The Optimal Option Investment Strategy team shall analyze call prices -5 to -50 in decrements of 5.

10.1.3.4 The Optimal Option Investment Strategy team shall incorporate stop-loss values ranging from 5 to 45 with increments of 5.

10.1.3.5 The Optimal Option Investment Strategy team shall analyze 45 days of trading a month.

10.1.3.6 The Optimal Option Investment Strategy team shall ignore any strategies missing more than 50% of trades.

10.2 Appendix 2: Project Plan

10.2.1 Introduction

Project Management of a project is an important aspect that needs to be carried out throughout the course of the project, to ensure that the project stays on schedule and within budget. This project's topic is of interest to the fields of operations research, specifically financial engineering, as well as systems engineering. Initially the project's scope is to perform a detailed analysis of the 6 years of historical data on Standard and Poor's 500 (S&P 500) options by applying straddle and strangle strategies to find an optimal range of investment opportunities. These high performing strategies would then be recommended to different users with different risk profiles. The second goal is to find the optimal budget allocation which is a tradeoff analysis between invest return and risk of catastrophic loss.

10.2.2 Work Breakdown Structure (WBS)

The goal of the WBS is to arrange all tasks in the project by task levels. Breaking down the project into small tasks facilitates the measurement and the control of the various activities on the project. This will also help the stakeholders to get a better idea of what needs to be accomplished in this project. Figure 26 below depicts the work breakdown structure for the Optimal Options Investment Strategy project.

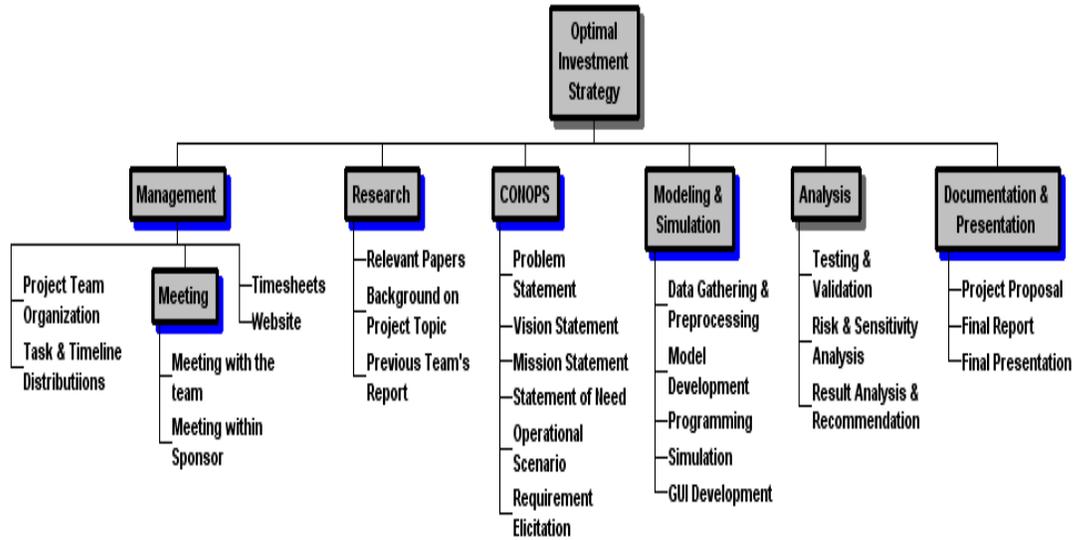


Figure 26

10.2.3 Individual Tasks

The team is comprised of five students Master’s students in Operations Research, with diverse backgrounds presented in Figure 27.

Team Member	Chen, Tony	Esmailzadeh, Ehsan	Jarvandi, Ali	Lin, Ning	O’Neil, Ryan
Background	Mathematics / Economics	Systems Engineering	Systems Engineering	Transportation Engineering	Computer Science

Figure 27

The individual tasks were defined for team members based on their educational background, professional background, interests, area of expertise, and required resources for this project. Figure 28 below shows high level responsibilities to each team member.

Tasks / Team Member	Chen, Tony	Esmailzadeh, Ehsan	Jarvandi, Ali	Lin, Ning	O’Neil, Ryan
Management	X	X	X		
Research	X	X	X	X	X

CONOPS		X		X	
Requirements Elicitation		X		X	
Design	X			X	X
Analysis	X	X		X	X
Data Parsing	X				X
Modeling	X				X
Programming					X
Testing & Validation	X	X		X	X
Presentation		X		X	
Documentation	X	X		X	X
Website					X

Figure 28

10.2.4 Project Budgeting and Earned Value Management

In order to effectively measure the project progress, we applied earned value management techniques to combine measurements of cost, schedule, and performance in a single integrated system. The team calculated the expected cost based on the schedule requirement from the course syllabus and multiplying by number of weeks, number of team members, and \$40 per hour cost of labor. Therefore, the forecasted budget came to \$32,000. This was only an estimate which changed depending on the team's performance. For example, the team considered an increase rate of productivity towards the end of the semester and decrease rate of productivity over the spring break. After assigning the resources to tasks, the team was able to produce the budgeted cost of work schedule for the entire course of the project as well as budgeted and actual cost of work performed for the entire course of project evolving over time. The actual cost at the end of the project came up to \$33,450. The team also produced the Cost Performance Index (CPI) and Schedule Performance Index (SPI) for the entire project, which is an indicator of the team's progress as a function of time.

Figure 29 shows the Earned Value Management (EVM) of the Investment Option Optimal Strategy team over the entire course of project. The blue line represents the budgeted cost of the work the team performed, which was estimated at the beginning of the project. The red line and the green line are indicator of the actual cost of the work performed and the actual values of the work performed respectively, evolving over the time.

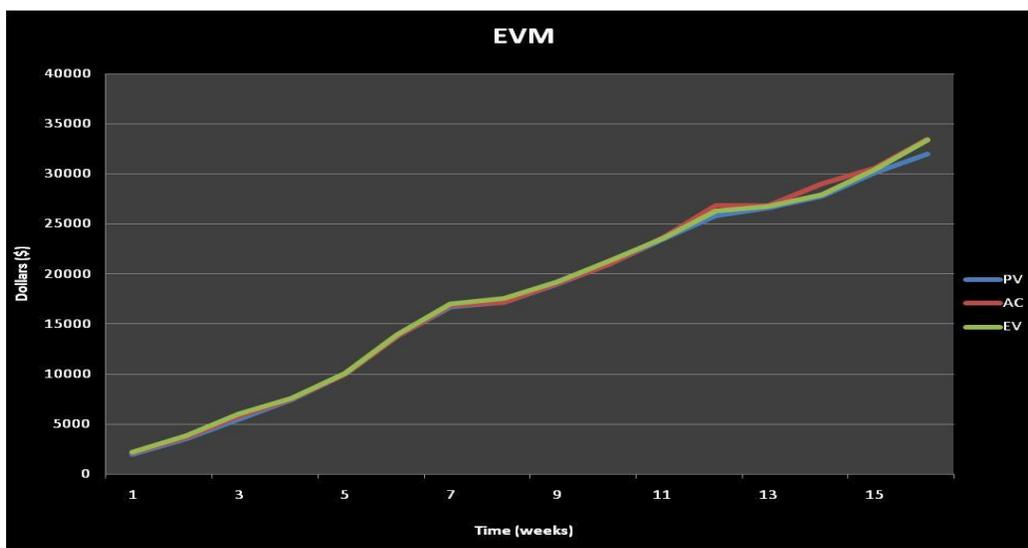


Figure 29

Figure 30 shows the Cost Performance Index (CPI) in blue and Schedule Performance Index in red. CPI is the ratio of earned value in terms of cost over the actual cost of the work, and SPI is the ratio of earned value in terms of cost over the budgeted cost of the work performed over the course of project. The goal is to keep CPI and SPI values greater than 1 to ensure that the team is performing on schedule and on budget.

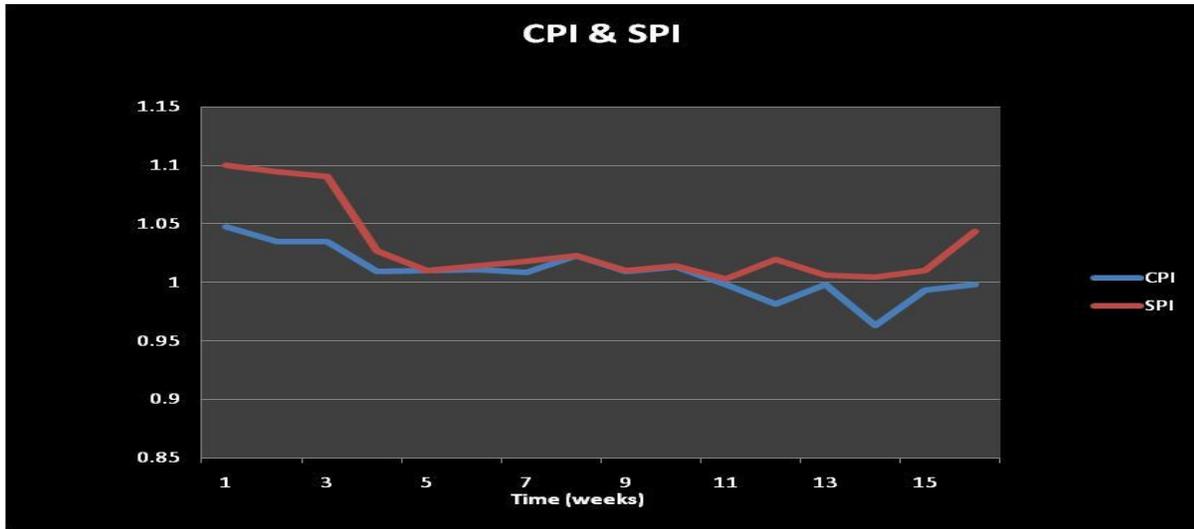


Figure 30

Figure 31 shows commits to our subversion source repository during the course of the project:

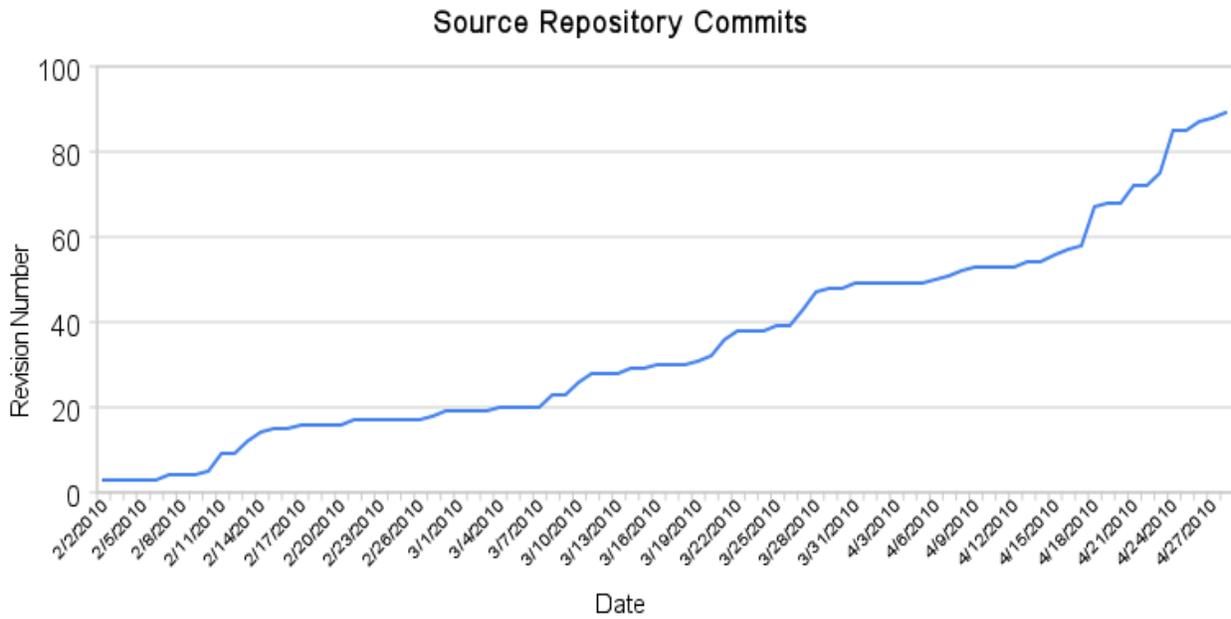


Figure 4

Figure 32 shows the total lines of Java code within the model over time. As of report date, the project consists of 3,329 lines of Java.

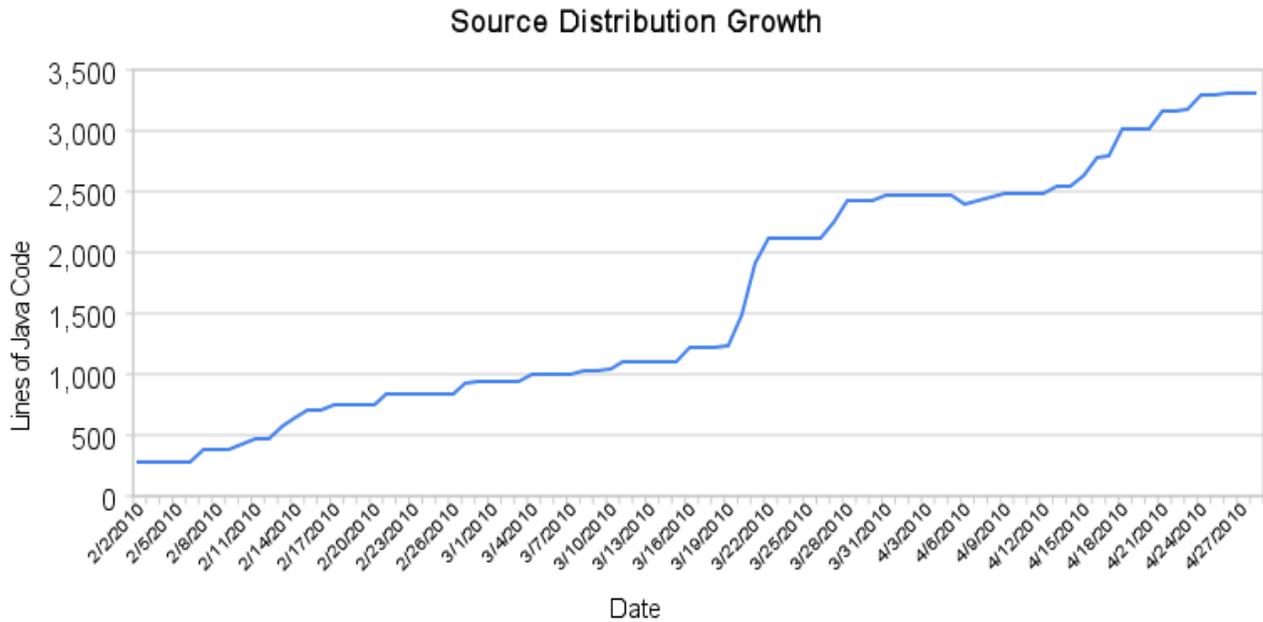


Figure 32

10.2.5 Project Development Schedule (Gantt chart)

To better illustrate the project the team used a Gantt chart to show duration, start and finish dates, and dependencies of each task. Each task is assigned to at least one member of the group depending on depth and work load. By using Microsoft Project the group is able to efficiently and accurately track the completion percentage of each task. This allows group members to be aware of the status of the project and see upcoming critical actions needed to achieve milestones. Figure 33 shows the Gantt chart for Optimal Option Investment Strategy project.

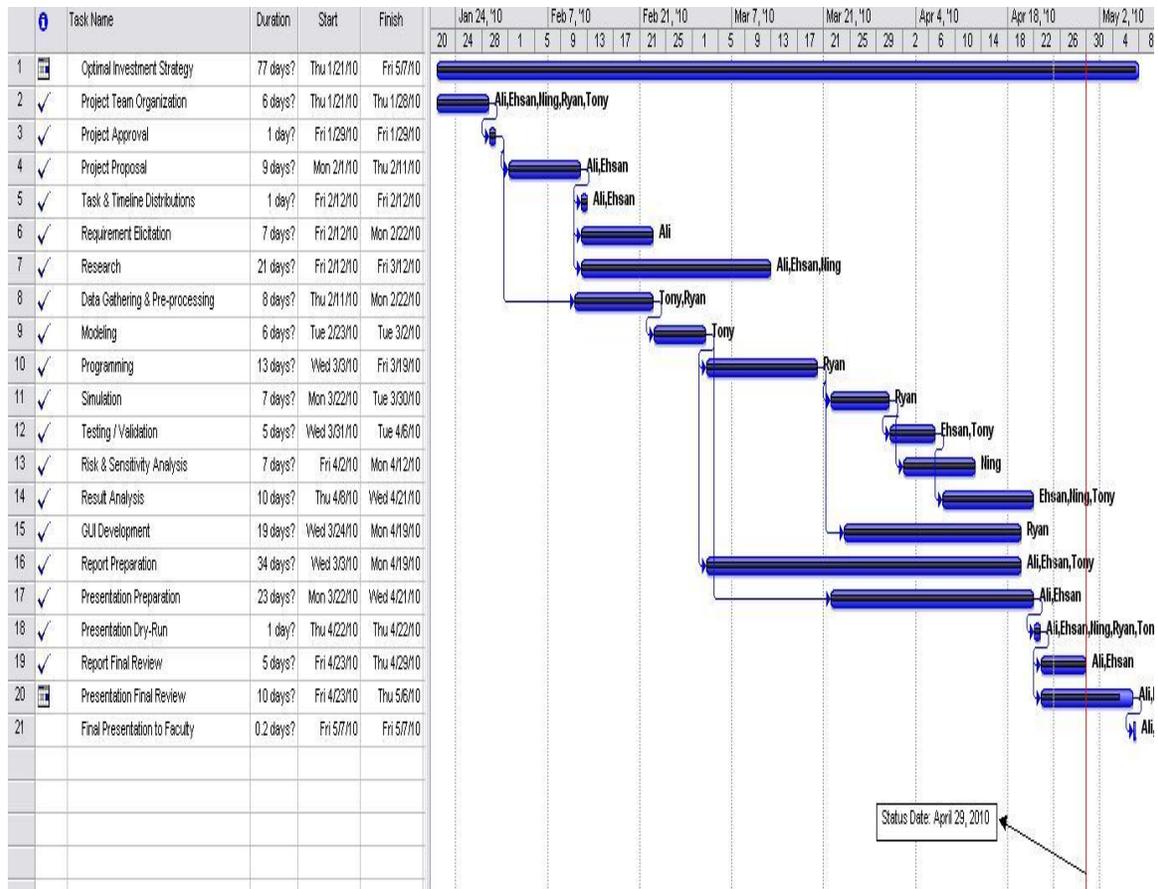


Figure 33

Deliverables

Deliverable	Delivery Date
Project Proposal	Feb 11, 2010
Status Report	Feb 18, 2010
Progress Report	March 4, 2010
Status Report	March 18, 2010
Progress Presentation	April 1, 2010
Final Report	April 29, 2010
Final Presentation	May 7, 2010

By the end of the semester the Optimal Investment Strategy team will also provide the project sponsor and the faculty with the following:

- Well documented executable model capable of providing optimal investment and allocation strategies for users with different risk profiles
- A user-friendly graphical application to provide the interface between the user and the model

10.3 Appendix 3: Strangles with Adjustments

The scope of this project was to study basic short strangles. More complex strategies were also implemented but their results were not studied. This strategy models a more realistic trading behavior that responds to changes in trader position.

Trade and forget strangles with stop loss and volatility combine to form over 100,000 strategies. In each strategy, the option writer sells a strangle every month and forgets it until expiration, unless it triggers a stop-loss. More likely, a trader will pay close attention to the market and be reactive to changes in position. If a stop-loss is triggered, the seller may try to recoup the loss incurred by that option by selling another one. Similarly, if a taken position is seen to be profitable before expiration, the seller may buy back the option for a profit and sell another that will generate the same premium as the original. These behaviors are called adjustments.

This type of behavior is modeled by updating short strangles with adjustments. Its fundamental components are the same as the basic short strangle described above, especially with regard to trading days and maximum volatility. There are some important differences:

1. Strike prices are not consistent indicators of premiums an option writer receives for selling options. Further, when using adjustments one desires to either recoup a known loss or sell a position with a known potential profit, each of which is done by looking at option premiums. Thus this model selects its options by premium instead of strike price. Specifically, it sells strangles with symmetric premiums on puts and calls of 5 through 25 in increments of 5.
2. Stop-loss on each option is treated as a factor of the original premium. If an option is sold with a premium of 10 and a stop-loss factor of 1.5, then that stop is triggered when the premium on the option surpasses 25. Upon hitting a stop-loss, the option writer sells a new option with the closest premium to the stop-loss value to recoup loss.
3. When the premium on an option dips to half its original value, the option writer buys back that option, earning the difference between the purchase price and the original premium. The strategy now sells another option with the original premium in order to capitalize more

on a good position. This is done through the end of the trading period, or 15 days prior to expiration. The adjustment algorithm is as follows:

1. Define start and end date of the trading period
2. Pick a stop-loss factor
3. Pick a maximum acceptable volatility value
4. Pick the next sequential expiration date
5. Pick a strategy (combination of call strike, put strike, trading date)
6. Check S&P 500 volatility of the trading date
 - a. If greater than maximum acceptable volatility, return to 5
 - b. If less than maximum acceptable volatility, record the premiums received from the two options and continue to 7
7. Process the next day between the trading date and the expiration date to check if stop-loss should be executed.
 - a. If an option hits a stop-loss, buy back the option and record the loss. Then sell another option with premium equal to the stop-loss. Return to step 7 and continue until expiration date.
 - b. If an option premium drops to half of what the options seller initially received, buy back the option and record the profit. Then sell another option of the same type with premium equal to the previous. Return to step 7 and continue until expiration date.
 - c. If no stop-losses are not executed, continue to step 8
8. Sum the gains and losses from the adjustment then add to the payoff at expiration.
9. Return to step 5 until all strategies have been processed
10. Return to step 4 until all expirations have been processed
11. Return to step 3 until all maximum acceptable volatilities have been processed
12. Return to step 2 until all stop-loss values have been processed

10.4 Appendix 4: Algorithms

10.4.1 Trade and Forget

1. Define start and end date of the trading period
2. Pick the next sequential expiration date
3. Pick a strategy (combination of call strike, put strike, trading date) and record the premiums received from the two options
4. Calculate and record the payoff from the difference between the premium prices at expiration and premium prices at trading date.
5. Return to step 3 until all strategies have been processed
6. Return to step 2 until all expirations have been processed.

10.4.2 Strangle with Stop-Loss

1. Define start and end date of the trading period
2. Pick a stop-loss value
3. Pick the next sequential expiration date
4. Pick a strategy (combination of call strike, put strike, trading date) and record the premiums received from the two options
5. Process all days between the trading date and the expiration date to check if stop-loss should be executed.
6. If both stop-losses are executed, take two losses at the stop-loss amount and return to step 4.
7. If both stop-losses are not executed, continue to step 9
8. If one stop-loss is executed, take one loss at the stop-loss amount and continue to step 9 to calculate the payoff for the other option
9. Calculate and record the payoff from the difference between the premium prices at expiration and premium prices at trading date.
10. Return to step 4 until all strategies have been processed
11. Return to step 3 until all expirations have been processed
12. Return to step 2 until all stop-loss values have been processed

10.4.3 Strangle with Volatility Driven Decision Making

1. Define start and end date of the trading period
2. Pick a stop-loss value
3. Pick a maximum acceptable volatility value
4. Pick the next sequential expiration date
5. Pick a strategy (combination of call strike, put strike, trading date)
6. Check S&P 500 volatility of the trading date
7. If greater than maximum acceptable volatility, return to 5
8. If less than maximum acceptable volatility, record the premiums received from the two options and continue to 7
9. Process all days between the trading date and the expiration date to check if stop-loss should be executed.
10. If both stop-losses are executed, take two losses at the stop-loss amount and return to step 5.
11. If both stop-losses are not executed, continue to step 13
12. If one stop-loss is executed, take one loss at the stop-loss amount and continue to step 13 to calculate the payoff for the other option
13. Calculate and record the payoff from the difference between the premium prices at expiration and premium prices at trading date.
14. Return to step 5 until all strategies have been processed
15. Return to step 4 until all expirations have been processed
16. Return to step 3 until all maximum acceptable volatilities have been processed.
17. Return to step 2 until all stop-loss values have been processed

10.5 Appendix 5: Pseudo Code

10.5.1 Optimal Fractional Investment

1. Setup initial amount $C = \$1,000,000$;
2. Let f = the fraction of money to invest in the market;
3. Let L = abs (the biggest point loss in our trades);
4. Let Margin = \$5,000
5. Let P = the points to earn or lose
6. For Strategy 1 to Strategy n
7. For $f = 0.05:0.05:1$ (this is MATLAB format which means 0.05 0.10 0.15...0.95 1)
8. NewMoney = C ;
9. For Trade 1 to Trade m
 - a. $B = \text{NewMoney} * f$;
 - b. $\text{TheNumberOfContract} = B / \max(L * 50, \text{Margin})$;
 - c. $\text{NewMoney} = \text{TheNumberOfContract} * P * 50 + \text{NewMoney}$;
 - d. $\text{TWR} = \text{NewMoney} / C$ (TWR should be displayed)
10. End

10.5.2 Modeling Risk of Ruin

1. To find a corresponding risk of ruin
 2. Setup initial amount $C = \$1,000,000$;
 3. Let f = the fraction of money we invest in the market;
 4. Let $L = \text{abs}$ (the biggest point loss in our trades);
 5. Let $\text{Margin} = \$5,000$
 6. Let P = the points we earn or lose
 7. Let $Z = 0.5$
 8. For Strategy 1 to Strategy n
 9. For $f = 0.05:0.05:1$ (this is MATLAB format which means 0.05 0.10 0.15...0.95 1)
 - i. $\text{NewMoney} = C$;
 - ii. For trade 1 to trade m
 1. $B = \text{NewMoney} * f$;
 2. $\text{TheNumberOfContract} = B / \max(L * 50, \text{Margin})$;
 3. $\text{Rate} = (\text{NewMoney} - \text{OldMoney}) / \text{OldMoney}$;
 4. $\text{NewMoney} = \text{TheNumberOfContract} * P * 50 + \text{NewMoney}$;
 5. $\text{TWR} = \text{NewMoney} / C$ (TWR should be displayed)
 - iii. End
 - iv. Get mean of rate of return = a ;
 - v. Get standard deviation of rate of return = d ;
 - vi. $R = e^{(-2 * a / d) * (\ln(1 - z) / \ln(1 - d))}$ (Risk of Ruin should be displayed)
 - b. End
10. End

10.6 Appendix 6: Sample Validation File

Generating return values for a strategy requires a large number of computations and data lookups. Below is an excerpt of these computations for a single strategy from the logging and debugging facilities of the model software. These can be turned on using the `DEBUG_STRATEGY` variable in the Settings class.

```
[ SHORT STRANGLE STRATEGY ]
DAYS BEFORE EXP: 42
PUT PRICE:      -15
CALL PRICE:     5
STOP-LOSS:     20
MAX VOLATILITY: null

EXPIRATION DATE:      Fri Jan 19 00:00:00 EST 2007
TRADING DATE:        Fri Dec 08 00:00:00 EST 2006
=====
PUT STRIKE PRICE:    1395
PUT PREMIUM (TRADE): 11.7
PUT STOP-LOSS ACTIVATED? NO. PROCEED WITH NORMAL STRANGLE LOGIC.
-----
PUT STRIKE PRICE:    1395
PUT PREMIUM (TRADE): 11.7
PUT PREMIUM (EXP):   0.0
PUT PAYOFF:         11.7
=====
CALL STRIKE PRICE:   1415
CALL PREMIUM (TRADE): 24.7
CALL STOP-LOSS ACTIVATED? NO. PROCEED WITH NORMAL STRANGLE LOGIC.
-----
CALL STRIKE PRICE:   1415
CALL PREMIUM (TRADE): 24.7
CALL PREMIUM (EXP):  21.7
CALL PAYOFF:        3.0
TOTAL PAYOFF:       14.7

EXPIRATION DATE:      Fri Feb 16 00:00:00 EST 2007
TRADING DATE:        Fri Jan 05 00:00:00 EST 2007
=====
PUT STRIKE PRICE:    1395
PUT PREMIUM (TRADE): 14.1
ESTIMATING EXP PREMIUM: ASSET=1458.9, STRIKE=1395: PREMIUM = MAX(0, 1395 - 1458.9) = 0.0
PUT STOP-LOSS ACTIVATED? NO. PROCEED WITH NORMAL STRANGLE LOGIC.
-----
PUT STRIKE PRICE:    1395
PUT PREMIUM (TRADE): 14.1
ESTIMATING EXP PREMIUM: ASSET=1458.9, STRIKE=1395: PREMIUM = MAX(0, 1395 - 1458.9) = 0.0
PUT PREMIUM (EXP):   0.0
PUT PAYOFF:         14.1
=====
CALL STRIKE PRICE:   1415
CALL PREMIUM (TRADE): 22.2
CALL STOP-LOSS ACTIVATED? YES: 2 DAYS BEFORE EXP, SETTLE = 43.8
CALL PAYOFF:        -20
TOTAL PAYOFF:       -5.9

EXPIRATION DATE:      Fri Mar 16 00:00:00 EST 2007
TRADING DATE:        Fri Feb 02 00:00:00 EST 2007
=====
PUT STRIKE PRICE:    1435
PUT PREMIUM (TRADE): 11.7
PUT STOP-LOSS ACTIVATED? YES: 17 DAYS BEFORE EXP, SETTLE = 44.7
PUT PAYOFF:         -20
=====
CALL STRIKE PRICE:   1450
```

```

CALL PREMIUM (TRADE):          19.7
CALL STOP-LOSS ACTIVATED? NO. PROCEED WITH NORMAL STRANGLE LOGIC.
-----
CALL STRIKE PRICE:             1450
CALL PREMIUM (TRADE):          19.7
CALL PREMIUM (EXP):            0.0
CALL PAYOFF:                   19.7
TOTAL PAYOFF:                  -0.30000000000000007

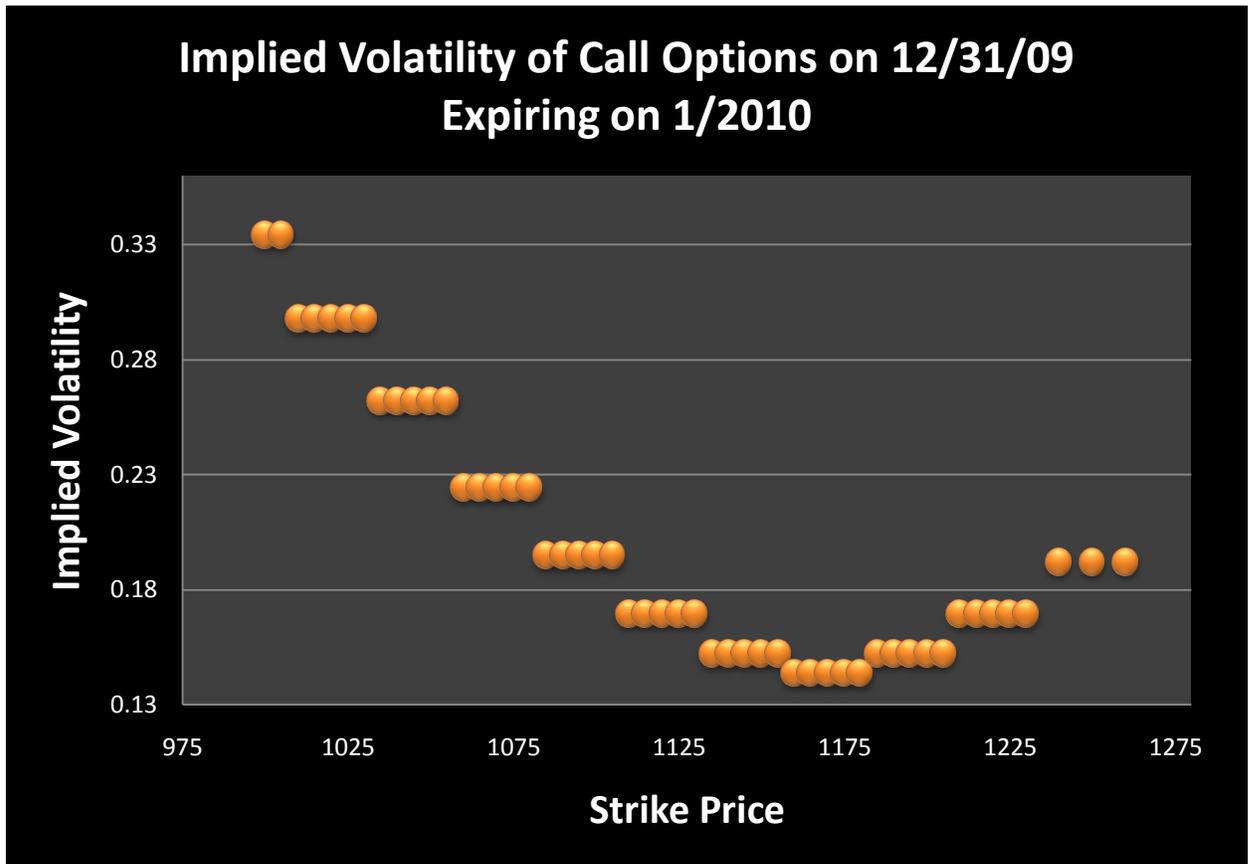
EXPIRATION DATE:               Fri Apr 20 00:00:00 EST 2007
TRADING DATE:                  Fri Mar 09 00:00:00 EST 2007
=====
PUT STRIKE PRICE:              1390
PUT PREMIUM (TRADE):           14.7
ESTIMATING EXP PREMIUM: ASSET=1493.1, STRIKE=1390: PREMIUM = MAX(0, 1390 - 1493.1) = 0.0
PUT STOP-LOSS ACTIVATED? NO. PROCEED WITH NORMAL STRANGLE LOGIC.
-----
PUT STRIKE PRICE:              1390
PUT PREMIUM (TRADE):           14.7
ESTIMATING EXP PREMIUM: ASSET=1493.1, STRIKE=1390: PREMIUM = MAX(0, 1390 - 1493.1) = 0.0
PUT PREMIUM (EXP):             0.0
PUT PAYOFF:                    14.7
=====
CALL STRIKE PRICE:             1405
CALL PREMIUM (TRADE):          31.2
CALL STOP-LOSS ACTIVATED? YES: 11 DAYS BEFORE EXP, SETTLE = 51.4
CALL PAYOFF:                   -20
TOTAL PAYOFF:                  -5.3000000000000001

EXPIRATION DATE:               Fri May 18 00:00:00 EDT 2007
TRADING DATE:                  Thu Apr 05 00:00:00 EST 2007
=====
PUT STRIKE PRICE:              1430
PUT PREMIUM (TRADE):           15.3
PUT STOP-LOSS ACTIVATED? NO. PROCEED WITH NORMAL STRANGLE LOGIC.
-----
PUT STRIKE PRICE:              1430
PUT PREMIUM (TRADE):           15.3
PUT PREMIUM (EXP):             0.0
PUT PAYOFF:                    15.3
=====
CALL STRIKE PRICE:             1445
CALL PREMIUM (TRADE):          27.5
CALL STOP-LOSS ACTIVATED? YES: 27 DAYS BEFORE EXP, SETTLE = 53.3
CALL PAYOFF:                   -20
TOTAL PAYOFF:                -4.6999999999999999

```

Figure 5

10.7 Appendix 7: Example of Volatility Smile



10.8 Appendix 8: Optimal Fraction and Risk of Ruin Formulas

10.8.1 Kelly Formula

The Kelly formula is used to determine the optimal size of a series of bets in order to maximize payoffs. A Kelly strategy will outperform other strategies in most gambling scenarios and some investing scenarios under some simplifying assumptions. (2008) *Kelly formula* finds:

$$\text{optimal fractional allocation: } f^* = \frac{b * p - q}{b}$$

Where:

b = odds of winning

p = probability of winning

$q = 1 - p$ = probability of losing

Acceptable level of risk and desired return are factors in determining the number contract to trade using fractional investment. According to John A. Anderson and Robert W. Faff, to maximize the geometric growth, one needs to identify the account capitalization required for each futures contract that produces the highest terminal wealth relative (TWR = Final balance/ Initial balance) to the original investment per futures contract for a range of f values. As the account capitalization per contract is a function of the f value selected and the largest loss, it is defined as (John A. Anderson 2004):

$$\text{Capitalization per contract} = \frac{\text{Largest Observed Loss}}{f} \text{ (for } 0 \leq f \leq 1)$$

If the largest observed loss is \$1000, then the optimal f aims to determine which amount of the capital should be applied per futures contract. At $f = 1$ (full allocation) the trader would invest \$1000 per contract but also face complete loss of capital if all contracts were to become worthless. Had the trader adopted a more conservative f value (that is $f < 1$) then extra capital per contract would be allocated and thereby preventing complete loss. For example, had a value of f

= 0.6 been adopted then the capitalization per contract is $\$1000/0.4 = \2500 and if the contracts became worthless we would still have \$3000 in capital (John A. Anderson 2004)

The number of futures contracts traded at a level of portfolio capitalization is a function of the optimum funding per contract divided by the account balance. Therefore, if the optimum funding per contract is \$10,000 and the trading account is \$100,000 then the trader would trade 10 futures contracts. Therefore the number of futures contracts to be traded is: (John A. Anderson 2004):

$$\text{Number of Contracts} = \frac{\text{Account Balance}}{\text{Capitalization per Contract}}$$

However, *Kelly's formula* is basing on two assumptions:

1. winning and losing per bet is constant;
2. the number of bets needs to be sufficiently large so that winning/losing percentages and average win/loss are accurate.

The first assumption does not fit our model and an approximation would introduce risk so we chose to model optimal fractional allocations.

10.8.2 Vince Formula

$$\sum_{M=1}^{\infty} \prod_{X=1}^M 1 - \frac{\left(\frac{1-P}{P}\right)^{\frac{Y}{A}} - 1}{\left[\left(\frac{1-P}{P}\right)^{\frac{(\bar{Q}+Q)-(1-Y)*Q}{Q*A}}\right] - 1} \quad \text{Where:}$$

$$Y = \frac{1}{X}$$

\bar{W} = average win

\bar{L} = average loss

p_w = probability of win

f = fractional investment allocation

$$Q = \left| \frac{-30000}{f} \right|$$

$$Z = \left| \frac{\bar{W}}{Q} \right| * p_w - \left| \frac{\bar{L}}{Q} \right| * (1 - p_w)$$

$$A = \sqrt{p_w * \left(\frac{\bar{W}}{Q}\right)^2 + (1 - p_w) * \left(\frac{\bar{L}}{Q}\right)^2}$$

$$P = \frac{\left(1 + \frac{Z}{A}\right)}{2}$$

It is possible that this formula returns values greater than one. Further, it assumes that trades are independent and outcomes of trades are stationary. Therefore we need another model to calculate risk of ruin for fractional trading. (Vince 1990) This model uses a formula which is widely found in the market. Most professional traders and hedge fund investors know the monthly mean and standard deviation of their returns. (Chamness 2009) These numbers are needed as inputs to the risk of ruin formula and can be measured over months, weeks, days, or trades. (Chamness 2009)

10.9 Appendix 9: Final TWR & Risk of Ruin Example Code

MATLAB code:

```
% input parameters;
% Q = 300000; % Q = abs(-30000/f). Here f = 0.1;
% Z = 0.00516733; % for case--f = 0.1;
% A = 0.050750997; % for case--f = 0.1;
% P = 0.550908688; % for case--f = 0.1;
AW = 13917;
PW = 0.6;
AL = -17000;
G = 0.5;
X = 2;
Y = 0.5;

for f = 0.1:0.05:1

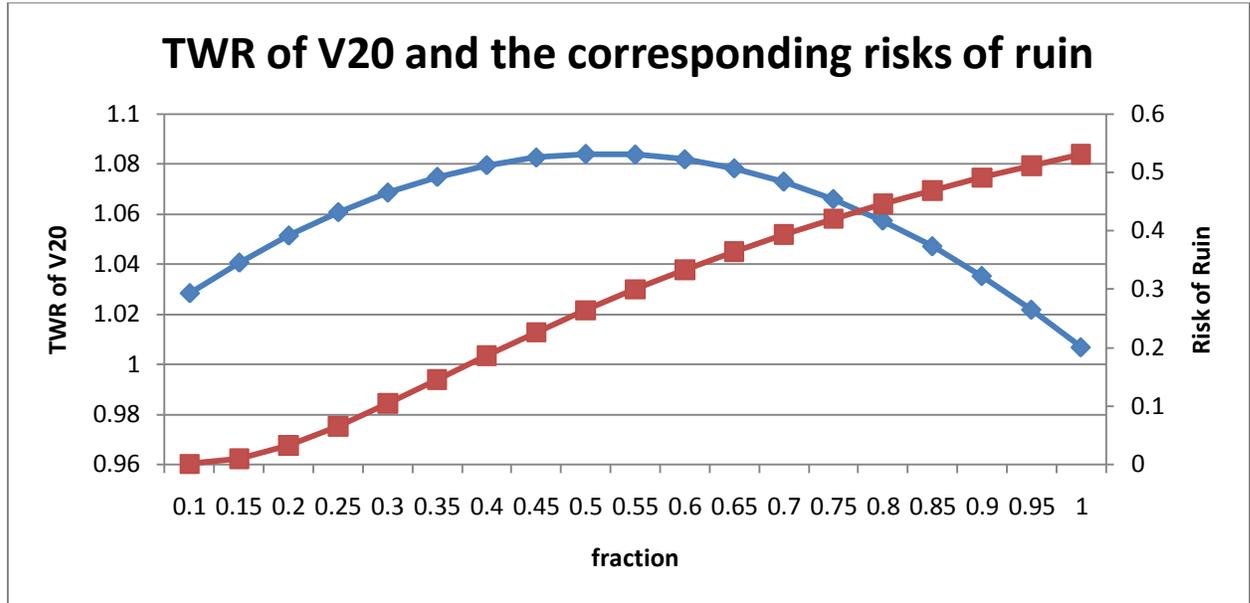
    Q = abs(-30000/f);
    Z = (abs(AW/Q)*PW)-(abs(AL/Q)*(1-PW));
    A = ((PW*(AW/Q)^2)+((1-PW)*(AL/Q)^2))^(1/2);
    P = 0.5*(1+(Z/A));

%for each f, calculate the final Risk of Ruin;
clear sample;
for i = 1 : 20
    sample(:,i)= 1 - (((1-P)/P)^(Y/A))-1)/((((1-P)/P)^((((Q/i)+Q)-((1-Y)*Q)/Q)/A))-1);
    %sample(:,i);% check sample(:,i);
    if i > 1
        New_sample(:,i)=prod(sample,2)+New_sample(:,i-1);
    else
        New_sample(:,i)=sample(:,i);
    end
    % We don't have to find the 20th one. This is a checker to see the
    % difference between Number (n) and Number (n-1);
    if (i > 1)
        if New_sample(:,i) - New_sample(:,i-1) < 10^(-6)
            break;
        end
    end
end
New_sample(:,i)
end

%sample' %to make it as a vector. We can check "New_sample" by this guy.;
%New_sample
```

TWR & Risk of Ruin Example Illustration

We calculate the profit and loss for each trade using the example data set as input. By utilizing the algorithm discussed above the final return for each selected fractional allocation value (red line) and associated risk of ruin (blue line) are shown in figure 42.



Blue line-final TWR, Red line-risk of ruin

10.10 Appendix 10: Investment Model Code

The software components of this project, including the investment model and GUI, consist of approximately 3,300 lines of Java code, spread across 30 different classes. Below is a breakdown of the more important classes.

`investmentallocation.Settings`

Contains configuration for the entire application, including input and output file locations, indexes for parsing data from strike price and asset price files, and strategy parameters such as days before expiration, put and call prices, stop-loss settings, and maximum volatility. It also allows one to turn on debugging output to validate the computations of the model.

`investmentallocation.InvestmentAllocationFractionOverTime`

`investmentallocation.InvestmentAllocationOptimalFraction`

`investmentallocation.InvestmentAllocationPlotWindow`

`investmentallocation.InvestmentAllocationReturnDataWindow`

`investmentallocation.InvestmentAllocationView`

These are the Swing classes that power the GUI. The graphical aspects are editable in the NetBeans GUI editor. The GUI application's main method lives in `InvestmentAllocationView`.

`investmentallocation.jobs.StrikePriceStoreJob`

Responsible for parsing input strike price and premium data and writing it to a binary file used by the model. This takes in all available option parameter data in a textual format and converts it to proper Java objects such as dates and floating point numbers. It computes calendar days until expiration for each option based on its trade date and the date of the third Friday of its expiration month. It also approximates call and put increments (+5, -5, and so on) based on the strike price of each option and the close of the S&P 500 on its trading day. All option data are indexed by expiration date, days before expiration, put offset and call offset and then written to a binary file known as a strike price store.

`investmentallocation.jobs.StrangleBinaryOutputJob`

Loads the strike prices from the strike price store and runs basic short strangles against them to determine return data. The returns and other computed metrics are written to a binary output file for use by the GUI. Values for the strangle parameters are set in the Settings class.

`investmentallocation.jobs.AdjustmentsBinaryOutputJob`

Follows the same logic as `StrangleBinaryOutputJob` but uses adjusted strangle strategies instead of basic short strangles.

`investmentallocation.options.S&P 500`

Provides a lookup table for S&P 500 index values by date. As input it takes a CSV file of S&P 500 adjusted closing values from Yahoo! Finance.

`investmentallocation.options.S&P 500 ExpPrices`

Provides a lookup table for S&P 500 futures index values by expiration year and month. The input data for this class was estimated based on the provided option strike price data.

`investmentallocation.options.VolatilityIndex`

Provides a lookup table for S&P 500 volatility index (VIX) values by date. As input it takes a CSV file of VIX adjusted closing values from Yahoo! Finance.

`investmentallocation.options.StrikePrice`

Encapsulates all the data needed to represent a strike price. This also calculates payoffs for an option given stop-loss value under basic short strangle and adjusted strangle strategies. These are generated by the `StrikePriceStoreJob` and put into a `StrikePriceStore`.

`investmentallocation.options.StrikePriceStore`

Provides a lookup table for strike prices by expiration date, days to expiration, put offset and call offset. This is created and written to disk by `StrikePriceStoreJob`.

`investmentallocation.strategy.Strategy`

An abstract base class that provides basic computations needed for strategy return metrics based

on actual option returns in points.

investmentallocation.strategy.Strangle

Responsible for modeling a strangle strategy over a given trading period. Requires strategy parameters from the StrangleBinaryOutputJob. Extends Strategy.

investmentallocation.strategy.AdjustedStrangle

Responsible for modeling an adjusted strangle strategy over a given trading period. Requires strategy parameters from the AdjustmentsBinaryOutputJob. Extends Strategy.

investmentallocation.strategy.InvestmentFraction

Computes metrics such as final TWR, average percent return, and risk of ruin for an instance of Strategy by computing actual returns for a given fraction of investment.

investmentallocation.output.OptimalFraction

Tests a given Strategy against each fraction of investment from 5% to 100% and stores the optimal fraction data based on highest final TWR.

investmentallocation.output.StrangleOutput

Encapsulates just the metrics required for showing a strategy in the GUI. Each instance contains the following fields:

- returnValues: list of returns in points for each trade. If a trade could not be accomplished, its corresponding return value is null.
- days: the number of days before expiration to trade in the given strategy.
- putDiff: difference between asset price and put strike price for this strategy.
- callDiff: difference between asset price and call strike price for this strategy.
- stopLoss: stop-loss value for this strategy. If it has no stop-loss value, this is null.
- maxVolatility: maximum volatility for this strategy. If it does not use volatility checks, this is null.
- numTrades: number of months in which trades are accomplished over the given time period using this strategy.
- avgReturn: average return per trade in points.
- avgPremium: average premium of the options sold, including both put and call.
- avgReturnOverPremium: average return divided by average premium.
- percentWins: percentage of time the given strategy sees positive returns.

- avgWin: average positive return in points.
- avgLoss: average negative return in points.
- stddevReturn: standard deviation of return in points.
- sharpeRatio: Sharpe Ratio (average return over standard deviation) for this strategy.
- TWRbestFraction: fraction of investment for this strategy that provides the highest final terminal wealth relative (TWR).
- TWRfinal: highest final TWR for this strategy.
- TWRworst: lowest TWR during the trading period for the fraction with highest final TWR.
- worstDrawDown: greatest percentage loss during the trading period for the fraction with highest final TWR.
- riskOfRuin: risk of ruin for the strategy with highest final TWR.
- avgRate: percentage return per trade accomplished.

investmentallocation.output.StrangleOutputStore

Provides a lookup table for StrangleOutput instances by strategy parameters. A file of this type is the output of StrangleBinaryOutputJob and AdjustmentsBinaryOutputJob. Files of this type are loaded by the GUI for filtering and display.

investmentallocation.util.DaysDifference

Computes the number of days between expiration and trading for an option.

investmentallocation.util.ExpirationDateCalculator

Computes the third Friday of a given year and month. The input option data only uses year and month for option expiration, so this is necessary to determining the number of calendar days between trade and expiration for each option.

Required Data Files and Libraries

The application requires input from a number of data sources in order to run. These include:

- End-of-day strike price data for S&P 500 futures options. This was purchased for use by the project sponsor. The data we have consists of thirteen CSV files from EEZEO001.csv to EEZEO013.csv.
- S&P 500 index value data from Yahoo! Finance. This can be downloaded in CSV format here: <http://finance.yahoo.com/q/hp?s=GSPC>
- S&P 500 volatility index data from Yahoo! Finance. This can be download in CSV format here: <http://finance.yahoo.com/q/hp?s=VIX+Historical+Prices>
- Interpolated expiration date values for the S&P 500 futures index. These were estimated from the options data and are available in the subversion repository for the software.

There are also third party free or open source Java libraries that are necessary for parsing input data and generating graphs. Library files for these are kept with the source code, so they need not be obtained separately. These include:

- JFreeChart: <http://www.jfree.org/jfreechart/>
- OpenCSV: <http://opencsv.sourceforge.net/>

Running the Application

There are three stages to running the application, each corresponding with a main method in the Java classes. The code is kept in a free private subversion repository at <http://www.xp-dev.com/> and is available by invitation. Upon obtaining a copy of the source code, a developer should assemble the required data files and place them on the file system in the INPUT_DIRECTORY from the Settings class. The developer should then run the main methods from these classes:

1. StrikePriceStoreJob: loads all the required input data into a form the Java application can use. The output of this job will be a file containing all strike price data called strikePrices.ser in the DATA_DIRECTORY location.
2. StrangleBinaryOutputJob: loads all the strike price data and runs all strangle strategies for different combinations of start and end year. Outputs various strangle output files in DATA_DIRECTORY.
3. AdjustmentsBinaryOutputJob: does the same as StrangleBinaryOutputJob for adjusted strangle strategies.
4. InvestmentAllocationView: launches the GUI application for browsing return data using whatever output file location is set in the setting STRANGLE_BINARY_OUTPUT.

The developer can also build a Java archive file (jar) for distributing the application. This is easy to do in the NetBeans or Eclipse integrated development environments (IDEs) or using the Ant tool. Once all the classes are all in an InvestmentAllocation.jar file, the user may run it against any of the data files generated in steps 2 and 3. For example:

```
java -jar InvestmentAllocation.jar strangleReturns-2007-2009-no-vix.ser
```

If the computer requires more memory for a particular file, that can be controlled by using the -Xms and -Xmx flags for Java. For example, the following command allocates 1000 MB of memory to run the application:

```
java -Xmx1000m -Xms1000m -jar InvestmentAllocation.jar strangleReturns-2007-2009-no-vix.ser
```

Appendix 11: Sample Raw Data

Sample raw data from Chicago Mercantile Exchange (C.M.E):

Trade Date	Ticker Symbol	Trade Hour (R/E)	Future/Option/Index Indicator	Delivery Month	Delivery Day	Contract Year	Strike Price	Open Ask/Bid Indicator	Opening Price	Opening Range Ask/Bid Indicator	Opening Range	High Ask/Bid Indicator	High Price	Low Ask/Bid Indicator	Low Price	Close Ask/Bid Indicator	Close Price	Closing Range Ask/Bid Indicator	Closing Range	Settle Cabinet	Settle Price	Actual Volume	Open Interest	Option Exercises	Implied Volatility	Exchange
70109	EZ	E	P	8	0	2009	965		0		0		0	A	52		0		0		58.6	0	66	0	0	CME
70109	EZ	E	P	8	0	2009	970		0		0		0	A	55.25		0		0		62.1	0	133	0	0.2105	CME
70109	EZ	E	P	8	0	2009	975		0		0		0	A	58.75		0		0		65.7	0	7	0	0	CME
70109	EZ	E	C	9	0	2009	630		0		0	B	289.25		0		0		0		290.5	0	3	0	0	CME
70109	EZ	E	C	9	0	2009	640		0		0	B	279.25		0		0		0		280.7	0	2	0	0.4188	CME
70109	EZ	E	C	9	0	2009	650		0		0	B	269.25		0		0		0		271	0	51	0	0.412	CME
70109	EZ	E	C	9	0	2009	690		0		0	B	230.5		0		0		0		232.3	0	50	0	0	CME
70109	EZ	E	C	9	0	2009	700		0		0	B	221.25		0		0		0		222.7	0	227	0	0	CME
70109	EZ	E	C	9	0	2009	710		0		0	B	211.75		0		0		0		213.1	0	13	0	0.3711	CME
70109	EZ	E	C	9	0	2009	720		0		0	B	202.25		0		0		0		203.7	0	367	0	0	CME
70109	EZ	E	C	9	0	2009	725		0		0	B	197.25		0		0		0		199	0	339	0	0	CME
70109	EZ	E	C	9	0	2009	730		0		0	B	192.5		0		0		0		194.3	0	431	0	0.3567	CME
70109	EZ	E	C	9	0	2009	740		0		0	B	182.75		0		0		0		184.9	0	68	0	0	CME
70109	EZ	E	C	9	0	2009	750		0		0	B	173.75		0		0		0		175.6	0	444	0	0	CME
70109	EZ	E	C	9	0	2009	760		0		0	B	164.25		0		0		0		166.4	0	99	0	0.3385	CME
70109	EZ	E	C	9	0	2009	770		0		0	B	155.5		0		0		0		157.3	0	355	0	0	CME
70109	EZ	E	C	9	0	2009	775		0		0	B	151		0		0		0		152.8	0	356	0	0	CME
70109	EZ	E	C	9	0	2009	780		0		0	B	146.5		0		0		0		148.3	0	371	0	0.3254	CME
70109	EZ	E	C	9	0	2009	790		0		0	B	146		0		0		0		139.6	0	78	0	0	CME
70109	EZ	E	C	9	0	2009	800		0		0	B	137.5		0		0		0		130.8	0	218	0	0	CME
70109	EZ	E	C	9	0	2009	805		0		0	B	132.75		0		0		0		126.5	0	2	0	0.3108	CME
70109	EZ	E	C	9	0	2009	810		0		0	B	128.75		0		0		0		122.2	0	331	0	0.3075	CME
70109	EZ	E	C	9	0	2009	815		0		0	B	124.25		0		0		0		118	0	1	0	0	CME
70109	EZ	E	C	9	0	2009	820		0		0	B	120		0		0		0		113.8	0	203	0	0	CME
70109	EZ	E	C	9	0	2009	825		0		0	B	115.5		0		0		0		109.7	0	17	0	0	CME
70109	EZ	E	C	9	0	2009	830		0		0	B	111.5		0		0		0		105.6	0	276	0	0.2966	CME
70109	EZ	E	C	9	0	2009	835		0		0	B	107.5		0		0		0		101.5	0	9	0	0.2939	CME
70109	EZ	E	C	9	0	2009	840		0		0	B	103		0		0		0		97.5	0	45	0	0	CME
70109	EZ	E	C	9	0	2009	845		0		0	B	99.25		0		0		0		93.6	0	385	0	0	CME
70109	EZ	E	C	9	0	2009	850		0		0	B	95.25		0		0		0		89.7	0	854	0	0	CME
70109	EZ	E	C	9	0	2009	855		0		0	B	91.25		0		0		0		85.9	0	4	0	0.2831	CME
70109	EZ	E	C	9	0	2009	860		0		0	B	87.25		0		0		0		82.1	0	15	0	0.2799	CME
70109	EZ	E	C	9	0	2009	865		0		0	B	83.5		0		0		0		78.4	0	70	0	0	CME
70109	EZ	E	C	9	0	2009	870		0		0	B	79.75	A	74.75		0		0		74.7	1	28	0	0	CME
70109	EZ	E	C	9	0	2009	875		0		0	B	76	A	71		0		0		71.1	1	1131	0	0	CME
70109	EZ	E	C	9	0	2009	880		0		0	B	72	A	67.5		0		0		67.6	5	321	0	0.2688	CME
70109	EZ	E	C	9	0	2009	885		0		0	B	68.75		0		0		0		64.2	0	3	0	0.266	CME

Number of options premiums by year from CME raw data (Rows marked in red were not analyzed)

Year	Available Options
1997	1974
1998	8130
1999	7002
2000	6371
2001	6352
2002	8415
2003	12461
2004	20848
2005	32207
2006	49468
2007	80900
2008	165955
2009	189765

10.11 Appendix 12: Output Data Summary

Average Final TWR 2007-2009

D.B.E	Average Final TWR
15	7.953378
16	9.985567
17	9.130557
18	4.485527
21	7.720662
22	10.00009
23	4.616528
24	5.256566
25	6.959904
28	6.129342
29	6.38973
30	3.704847
31	13.1059
32	7.96219
35	23.3305
36	23.86886
37	9.183648
38	25.83348
39	52.81807
42	61.59855
43	33.90807
44	70.72044
45	19.0903
46	13.3704
49	17.58188
50	9.763464
51	5.820596
52	5.831918
53	8.410436
56	4.605684
57	4.764026
58	3.876945
59	16.57618

Average Final TWR 2004-2006

D.B.E	Average Final TWR
15	2.73
16	2.83
17	2.60
18	1.80
21	2.32
22	2.29
23	4.23
24	3.73
25	3.82
28	3.32
29	4.79
30	5.67
31	2.76
32	1.80
35	3.46
36	3.25
37	3.04
38	3.08
39	2.59
42	2.51
43	3.04
44	3.05
45	2.96
46	2.55
49	2.88
50	2.93
51	2.88
52	3.79
53	3.60
56	2.49
57	2.88
58	5.31
59	3.67

Average Final TWR different Stop-Loss values 2007-2009

Rows are days before expiration and columns are stop-loss values

	5	10	15	20	25	30	35	40	45	None
15	8.06	7.9	11.01	9.39	8.96	10	8.66	7.49	6.66	1.4
16	5.36	4.87	11.07	13.42	17.04	15.82	13.86	10.34	6.98	1.1
17	7.28	6.94	8.26	11.74	13.76	13.85	10.45	9.24	8.68	1.1
18	8.83	4.33	2.18	3.19	5.41	5.41	5.31	4.89	4.27	1.03
21	8.6	7.43	7.27	8.85	10.82	11.28	10.26	9.22	8.75	1.1
22	4.59	6.15	7.03	8.67	8.07	5.53	4.61	4.41	4.25	1.01
23	9.53	6.03	4.94	4.3	5.24	5.34	3.81	3.06	2.8	1.1
24	5.26	2.8	5.06	9.67	9.81	7.23	4.15	3.32	3.62	1.65
25	1.71	2.05	6.9	13.43	13.71	12.04	9.07	5.78	3.71	1.21
28	4.52	4.68	5.94	8.42	9.5	8.34	6.51	6.09	5.96	1.32
29	9.36	7.36	7.43	8.92	7.32	5.02	5.33	6.81	5.15	1.2
30	10.76	9.43	6.05	2.43	1.45	1.17	1.15	1.19	1.59	1.82
31	18.14	21.57	17.76	29.04	16.48	11.59	7.17	5.15	2.95	1.22
32	20.57	20.84	13.49	8.17	5.64	3.87	2.63	1.96	1.46	0.99
35	60.93	38.44	33.81	34.3	26.09	16.93	10.87	6.77	4.05	1.12
36	46.2	29.37	28.8	56.11	37.17	18.99	10.76	6.82	3.36	1.11
37	25.4	16.39	15.59	14.04	5.53	4.88	3.61	2.68	2.5	1.2
38	39.53	59.6	44.84	36.79	28.59	17.93	13.78	9.11	7.04	1.13
39	40.11	87.73	132.79	113.92	77.74	41.72	20.32	9	3.91	0.95
42	18.58	42.28	179.8	210.3	96.11	41.46	15.8	6.94	3.6	1.11
43	10.17	12.08	60.7	75.66	65.94	63.04	28.88	13.59	7.94	1.09
44	12.42	26.89	169.89	151.53	141.43	89.01	65.31	30.57	18.98	1.17
45	46.77	28.55	20.9	13.37	16.53	20.29	20.07	14.62	8.79	0.99
46	27.24	18.57	14.68	9.58	15.31	22.63	22.82	16.79	13.42	1.27
49	21.97	9.44	7.59	11.07	33.36	37.18	23.09	13.9	17.21	1.01
50	8.46	7.44	11.15	17.86	15.09	9.66	8.79	8.39	9.79	1.01
51	5.53	3.55	3.06	4.64	10.28	12.65	7.55	6.29	3.65	1
52	1	0.97	1.84	12.92	18.49	8.33	4.44	3.67	5.45	1.21
53	0.99	0.93	1.04	1.68	4.82	13.19	19.5	20.15	20.15	1.66
56	4.16	3.09	2.07	3.14	3.32	2.88	4.53	6	15.71	1.15
57	2.78	1.67	6.27	11.4	6.79	4.02	2.84	3.94	6.83	1.09
58	1.77	4.76	6.53	8.62	4.68	4.27	2.51	1.9	2.45	1.3
59	19.12	17.12	41.65	32.8	16.83	10.31	12.68	8.69	5.4	1.18
60	19.61	12.01	20.95	33.99	20.97	16.17	11.09	13.25	10.29	0.99

Average Final TWR different Stop-Loss values 2004-2006

Rows are days before expiration and columns are stop-loss values

	5	10	15	20	25	30	35	40	45	None
15	2.54	2.92	2.85	2.71	2.53	2.57	2.59	2.68	2.85	3.04
16	2.2	2.28	2.51	2.5	2.8	3.05	3.24	3.24	3.24	3.24
17	2.35	2.14	2.25	2.58	2.68	2.69	2.79	2.85	2.85	2.85
18	2.26	1.68	1.55	1.57	1.66	1.76	1.82	1.89	1.89	1.89
21	2.49	2.04	1.87	1.93	2.1	2.32	2.49	2.61	2.7	2.7
22	2.57	2.17	2.04	2.17	2.39	2.33	2.38	2.33	2.28	2.25
23	2.33	2.94	3.43	4.5	4.9	4.71	4.78	4.86	4.9	4.9
24	2.56	2.95	3.35	3.9	4.13	4.06	4.02	4.02	4.09	4.18
25	2.74	3.76	4.16	4.65	4.2	3.99	3.8	3.69	3.6	3.66
28	3.54	4.44	4.47	3.94	3.35	3	2.78	2.61	2.51	2.54
29	3.8	5.59	5.95	6.08	5.33	4.64	4.38	4.09	3.94	4.05
30	3.33	5.55	6.63	6.56	6.1	6.12	5.8	5.59	5.45	5.59
31	5.24	3.69	3.07	2.74	2.43	2.21	2.03	2	1.99	2.25
32	3.15	2.6	2.12	1.89	1.52	1.36	1.3	1.32	1.31	1.39
35	4.9	5.18	5	4.21	3.32	2.62	2.31	2.17	2.32	2.58
36	6.27	5.74	4.6	3.5	2.85	2.17	1.9	1.74	1.71	2.01
37	4.06	3.8	3.45	3.3	2.8	2.73	2.84	2.56	2.4	2.49
38	4.65	3.98	3.06	2.85	2.99	2.98	2.73	2.57	2.38	2.57
39	2.81	2.82	3.37	3.04	2.75	2.44	2.38	2.19	2.03	2.07
42	3.67	3.51	3.53	3.13	2.34	2.02	1.86	1.83	1.64	1.59
43	2.89	3.41	3.3	3.57	3.19	3.16	2.82	2.65	2.53	2.87
44	2.45	2.46	2.77	3.37	3.59	3.46	3.16	2.93	2.98	3.31
45	2.56	2.62	2.59	3.51	3.87	3.3	2.98	2.76	2.57	2.87
46	2.28	2.03	2.26	2.8	2.44	3.11	2.84	2.49	2.41	2.8
49	3.39	3.51	3.47	3.48	3.26	3.07	2.94	2.83	2.77	2.91
50	2.89	3.23	2.65	2.83	2.81	3.14	3.11	2.9	2.82	2.89
51	1.87	3.18	3.06	3.28	3.36	3.14	2.8	2.64	2.56	2.9
52	2.44	2.6	3.29	4.63	4.79	4.6	4.1	3.71	3.57	4.17
53	3.33	4.23	3.75	4.79	4.17	3.6	3.33	3.11	3	2.71
56	4.37	3.94	3.09	2.8	2.59	2.08	1.8	1.45	1.37	1.39
57	3.32	3.96	4.88	4.18	3.27	2.77	2.04	1.6	1.34	1.43
58	1.88	4.93	8.42	8.47	6.55	5.26	4.68	4.22	3.97	4.73
59	2.8	2.8	5.61	6.01	4.91	3.8	3.16	2.74	2.52	2.38

Average premium by year

Rows are days before expiration and columns are years

	2005	2006	2007	2008	2009
60	24.89	31.55	58.03	79.08	69.62
59	26.11	30.41	46.27	80.65	73.38
58	24.43	29.04	45.73	79.54	68.80
57	27.20	28.91	47.59	78.20	64.99
56	26.81	28.47	45.06	78.33	69.05
53	24.91	27.81	49.91	78.33	68.05
52	23.99	27.14	47.47	74.91	64.58
51	22.31	25.78	42.78	72.79	63.32
50	23.19	24.96	44.67	73.18	57.06
49	22.78	23.92	45.48	71.78	63.77
46	19.92	24.64	43.21	72.13	61.16
45	21.99	24.42	41.03	66.93	59.91
44	20.08	23.95	41.95	67.68	59.21
43	20.40	24.04	42.19	66.04	58.24
42	18.90	22.52	42.07	68.17	56.01
39	18.29	21.40	41.48	65.58	53.79
38	18.33	21.35	39.08	63.56	51.64
37	17.79	20.85	38.62	63.54	50.74
36	17.61	20.03	38.46	61.64	47.79
35	16.21	19.76	36.03	62.58	46.20
32	15.13	18.82	37.54	60.46	41.53
31	15.04	17.62	31.04	56.66	43.85
30	13.95	15.77	29.55	58.31	41.49
29	14.15	15.88	25.99	53.45	39.77
28	13.03	15.61	28.77	53.66	37.94
25	12.30	15.24	31.73	52.31	36.34
24	11.73	13.85	27.33	48.36	34.00
23	10.95	12.94	24.78	45.29	32.82
22	10.93	12.08	25.21	42.13	31.07
21	10.23	10.83	23.91	43.18	31.74
18	9.77	10.01	22.22	43.90	29.55
17	9.01	10.19	22.65	38.31	26.98
16	8.05	9.17	19.81	37.35	25.16
15	7.64	8.77	18.89	37.82	24.21

Average Monthly Percentage Return, 2004-2006 and 2007-2009

Rows are days before expiration and columns are years

	2007-2009	2004-2006
15	5.78	3.46
16	10.70	3.42
17	6.91	3.14
18	6.79	2.15
21	7.69	2.91
22	5.15	3.13
23	5.13	4.62
24	6.09	4.19
25	12.99	4.76
28	6.13	4.20
29	8.62	5.42
30	9.55	6.12
31	12.26	4.06
32	8.93	2.07
35	7.71	4.92
36	12.62	4.48
37	10.05	4.78
38	8.26	4.73
39	8.99	4.01
42	11.74	4.20
43	8.63	5.35
44	7.20	5.12
45	11.36	5.86
46	9.22	5.99
49	7.02	5.75
50	8.82	6.06
51	9.67	5.86
52	8.17	7.59
53	6.56	7.50
56	9.66	5.42
57	7.51	6.31
58	4.65	8.72
59	4.21	7.01
60	6.28	

Maximum Acceptable Volatility

Rows are final TWR buckets and columns are maximum acceptable volatility

	30	50	None
TWR<1	1507	1865	1053
1<TWR<5	20410	14306	12635
5<TWR<20	10627	14933	14091
20<TWR<50	722	2312	4226
50<TWR<100	53	482	1249
TWR>100	1	102	746