UCHICAGO

TRADING COMPETITION

2022

Participant Case Packet

THE UNIVERSITY OF CHICAGO | Career Advancement Financial Markets

The University of Chicago Financial Markets Program (FM) is excited to show its in-house trading platform χ-Change built by senior members of the FM program for the 10th Annual UChicago Trading Competition! Cases 1 and 2 for the competition will be run live utilizing this platform. Case 3 will be run before the competition and results will be played back at the event.

In the next few weeks, you will receive emails detailing instructions on accessing the trading platform and practice data and on accessing Piazza, which will be used to address competitor questions and provide important case and platform updates.

Algorithm Development

Competitors may develop their algorithms in any computing language; however, Python will be the only officially supported language. No other languages will receive explicit support from the case writing team. On the day of the competition, one user from each team will be responsible for manually starting the team's algo at the beginning of each case round.

Additional details on rules and requirements for each round can be found in the case descriptions.

Case Submission Dates

For Cases 1 and 2, all competitors must submit a draft of their code by **noon (12:00 PM CST) on Friday, April 1st**.

Competitors will run their finalized algorithms locally on the day of the competition.

For Case 3, competitors' final algorithms must be submitted by **noon (12:00 PM CST) on Friday, April 1st**.

This case will be run in advance of the competition. Teams will not run their algorithms live on the day of the competition. Final scores will be announced on the day of the competition.

Table of Contents

CASE 1: Market Making

Introduction

Lumber is wood that has been processed into beams and planks. It is commonly used as a raw input for construction, furniture making, and paper making. Unlike most other commodities, lumber can be harvested throughout the year. However, the operational conditions for logging tend to be dependent on seasonal temperatures throughout the year. In addition, heavy precipitation often makes it more difficult to harvest lumber since operating conditions become more dangerous. For the above reasons, businesses who use lumber as a raw input often choose to hedge their risks using futures contracts.

In this case, you have been tasked to act as designated market makers for the lumber futures market. To avoid the hassle associated with the physical delivery of lumber, your firm has not permitted you to directly trade in the lumber spot market. Instead, you have been provided with historical daily spot prices and monthly precipitation amounts to assist you in coming up with a successful strategy to make markets on the lumber futures market.

Case Specifications

Each year will have a total of 252 trading days, with 21 trading days per month. At the start of each month, you will receive a forward-looking forecast of the next month's precipitation. For the purposes of this case, you may assume that the forecast is completely accurate.

Note that you will **NOT** be able to trade in the spot market nor have access to spot prices on the actual day of the competition.

For simplicity, each futures contract will be assumed to only trade 1 unit of the underlying. The contract for that month will expire on the last reading day of the month. For example, the June contract will expire on the 126th trading day of the year. Contracts at expiration will be cash settled and market participants will have the ability to trade contracts up until the day of expiration. The tick size is 0.01 for all contracts.

The codes for the future contracts are given below:

Expiration Month	Contract Code
April	LBSJ
June	LBSM
August	LBSQ
October	LBSV
December	LBSZ

Round Specifications

There will be 5 rounds corresponding to markets in the following years: 2022(practice), 2022, 2023, 2024, 2025 (you will have data from 2016-2021. The chronology is not irrelevant to succeeding in this case).

The 2022 data will be generated independently between the first and second runs: the former will serve as practice for familiarity with the server and model parameters.

Positions and PnL are not carried over between rounds.

Data and platform simulation will be provided so that you can train your algorithm prior to the competition.

Rules

You may take long or short positions in all futures available in a round, subject to risk limits specified below. There is a maximum order size of 100 in this competition. Exceeding it will result in the rejection of your entire order.

Risk Limits

As a market maker, your firm stipulates that the maximum long or short position you may hold in any one contract is 100.

If you exceed this position, all further orders for that contract that exacerbate the risk limits will be blocked by the exchange.

Scoring

At the conclusion of each round, each team will receive points corresponding to their rank.

The team with the highest PnL will receive 1 point, the team with the second highest PnL will receive 2 points, and so forth.

Any team disqualified during a round will receive points calculated as (# of Teams + 1).

The lowest point total wins. Ties will be broken by final round rank.

Note that the practice round will not count towards total points.

Case Materials & Data

Python stub code and training data will be released with the case packet through the UChicago Trading Competition Piazza. The basic bot will outline some key functions and exchange syntax for you.

Code Submission

We are requiring a preliminary **code submission by noon (12:00 PM CST) on Friday, April 1st.**

Competitors will however run their bots from their own computers on the actual day of the competition.

Miscellaneous Tips

In order to be successful in this case (and as a market maker in general), you will need:

1. A model that predicts the "fair value" of the asset you are trading
2. A market making algorithm that uses the predicted "fair value" to quote bids and asks while managing risk

Some possible points of consideration are:

- How to best make use of the historical data you are provided with to make predictions on unseen data?
- What modifications (if any) should you make in the way you trade if the fair price predicted by your model deviates from the mid-price of the market? What if only one side of your quotes gets filled?

- Would it make a difference for you to hold contracts till expiration, or would it make more sense to clear your positions before that?
- Is it always reasonable to quote symmetrically (in size and/or price) about your predicted fair price?
- Are there any circumstances where it would be justified to "cross the spread" as a market maker?

Note that as the exchange will fill orders on price/time priority, the speed of your market making algorithm matters. Write lines of code that submit orders more quickly and speed up the bot.

Questions

For questions regarding Case 1, please post in the UChicago Trading Competition Piazza in the "case1" folder. We will regularly check for new messages.

CASE 2: Options Trading

Definitions:

- Option: A financial derivative that gives the owner the right to buy or sell an asset at a pre-specified price at a pre-specified time (This case deals with European Options only. The definition of an American Option is slightly different)
- Exercise: Use the right granted by an option to buy or sell an asset
- Expiration: When the owner of an options contract must either exercise or give up. After this the contract ends.
- Strike Price: The pre-specified price at which the owner of an option can buy or sell the underlying asset
- Call Option: An option that gives the owner the right to buy
- Put Option: An option that gives the owner the right to sell
- Premium: The price of an option
- Volatility: The annualized standard deviations of returns of an asset. It is the only unknown input into basic option pricing formulas, so options trading is often thought in terms of volatility.
- Implied Volatility: The level of volatility implied by options prices
- Realized Volatility: The level of volatility an asset has historically experienced
- In The Money (ITM): An option which could be exercised for a profit
- At The Money (ATM): The option whose strike best matches the price of the underlying
- Out of The Money (OTM): An option which could not be exercised for a profit
- Intrinsic Value: The value of an option if it were exercised now
- Greeks: The risk measures typically used in options trading. Delta, Theta, Vega, and Rho measure the derivative of the price of an option relative to the price of the underlying asset, time (measured in years), volatility, and interest rates respectively. Gamma measures the second derivative of the price of an option with respect to price of the underlying.

Use Cases for Stock Options

Options can help both hedgers and speculators take positions and are a useful tool to anybody who wants a return profile that is non-linear with respect to an underlying asset.

For example, options can help pension plans ensure that they will be able to meet their payment obligations. If a pension fund has enough money to meet its payment obligations to pensioners and has enough of a surplus to be able to survive a 10% loss, but wants to generate additional returns, the pension fund might purchase a stock and a put option with a strike price 10% down from the current price. That way, they can make money if the stock goes up but also don't become insolvent if the stock falls more than 10%.

Options can help speculators as well by allowing them to bet more heavily on a certain price move than they would otherwise be able to. For example, if a stock costs $100, the 1 month call option with a strike at $100 costs $1, and a speculator thinks the stock will go up $22 in the next month, the speculator could either buy the stock for $100 or buy two 1 month calls struck at $110 for $1 each. If the speculator is correct and they bought the stock, then they would make $22 while deploying $100 in capital. If they bought 2 options they can exercise them at expiration, buying two shares of stock for $110 each, which they can immediately sell for $122 each, netting them $22 in profit (after subtracting the option premium) with only $2 of capital deployed. With $100 of capital deployed this would net $1100.

Options Pricing Models

Options can be priced using the Black Scholes Model or using a tree model. Both models take the interest rate, the price of the underlying asset, the time to expiration, the strike price, and the implied volatility of the underlying asset as inputs. All of these are known except for the implied volatility, which can be backed out from the price of other options on the market. However, the overall level of volatility is not known, and is often a reflection of market sentiment. As a result, volatility often forms in "clusters".

Both models work by assuming risk-neutrality, constant volatility, a lognormal distribution of underlying returns. However, these assumptions are not often true in practice, which is why options have "implied" volatilities. Generally, the closer the underlying is to the option price, the lower the implied volatility, which greater a volatility "smile".

Case Description

For this case, you will trade options on 5-10 simulations of an underlying. Each "run" will have a total of 1000 underlying updates. There will be a total of 5 option contracts, from those almost always ITM to those almost always OTM. The underlying will start at "100", and the options will be struck in increments of 10 from 80 to 120 (both puts and calls). You will have to create a dynamic algorithm which helps you value parameters that factor into option prices, and then quote prices based on your algorithm. The underlying represents a relatively illiquid index, which is subject to macroeconomic shocks, as well as sentiments about the index itself at a given point in time. There will be three events present during the case: a global pandemic scare, market uncertainty, and a speculative bubble. You will either need your algorithm to account for these cases or change your parameters/strategy when each event occurs. You will not be able to change the parameters in real time, but the occurrence of the event, which will be "announced" by the exchange, can "trigger" a change in your algorithm. Options will expire beyond the time horizon, so you won't be able to "cash out" as part of your strategy.
Limit and market orders will be allowed in options, and only market orders will be allowed in equities. Teams have the choice to build their algorithms using whichever programming language that implements gRPC binding; however, Python will be the officially supported language. No other languages will receive explicit support from the case writers.

While learning how to price options is an important part of this case, you'll also have to be an efficient trader to profit from any pricing advantage that you might have. This said, an important task of yours is to synthesize market data to be able to quote orders that are both executable and offer positive expected profit.

Risk Limits

Your risk limits for each stock and associated options chain are as follows. Any relationships between stocks are not considered when assessing risk.

Delta	2000
Gamma	5000
Vega*	1000000
Theta	5000

*Volatility is typically quoted in vol points, which are hundredths of mathematical volatility (the annualized standard deviation of returns). This Vega number is with respect to mathematical volatility, not vol points. If implied volatility increases from .22 to .23 you're permitted to gain or lose up to $10,000. If implied volatility increases from .22 to 1.22 you're permitted to gain or lose up to $1,000,000.

We do not have a risk limit for Rho because interest rates are fixed at 0 for this case.

Competitors who trade through the risk limits will be auto liquidated by the exchange until they are within risk limits. No consideration will be given to the most efficient way to decrease the risk of your portfolio and forced liquidation will be triggered for all underlying assets if any of them are outside of the risk limit. This will be very costly and an inefficient way to decrease risk so try to manage your risk yourself. This will also be exploitable if you're able to trade against price insensitive competitors who are being liquidated.

Tips and Hints

- After you've built a working options pricing model, you will need a value for implied volatility to price options on an underlying asset. Perhaps start with the realized volatility for each asset to get an initial estimate of what the implied volatility level for each stock should be. What other models are used to model volatility in practice?
- Consider how each event affects the underlying price and volatility – see if you can use your deductions to build "triggers" for your option prices. What option strategies may be good to implement so your trades best reflect your hypotheses about the market? For example, how can you long volatility without betting on the price movement?
- Events aside, consider how sensitive is each option to movements in underlying price versus volatility.
- Remember, the goal here is to profit relative to other competitors, not necessarily to be able to quote the "perfect" price. When events occur, how should you respond based on how other competitors are trading? Both profits and consistency matter, so see if there are any ways to increase your consistency, even if it means less profit on an individual time-period. Remember that you have multiple runs, each with a similar "story".
- Consider the effects of each of the Greeks on your prices. When do some Greeks play more of a part than others? Are they only dependent on what effect they are reflecting (for example, is delta only dependent on S?)?
- Be careful with risk limits. You can add the risk of all assets in your portfolio to get the risk of your entire portfolio. If you own a call option struck at 100, with .5 Delta, .03 Gamma, 10 Theta, and 90 Vega, a put option struck at 95 with -.05 Delta, .005 Gamma, 3 Theta, and 25 Vega, and a short share of stock, your portfolio's overall risk is -.55 Delta, .035 Gamma, 13 Theta, and 115 Vega. Your Delta position is -.55, not .45, because the short share of stock has a Delta of -1 (as the price of the stock goes up a dollar the short position loses a dollar). A simple way to calculate risk is by seeing how much the price moves when the risk-factor moves. For example, for some small, ε, we can compute a given option's Delta as follows: $\frac{\partial C}{\partial S} = \frac{1}{\varepsilon} C[(S + \varepsilon, \mathrm{K}, \mathrm{T}, \mathrm{r}, \sigma) - (S, \mathrm{K}, \mathrm{T}, \mathrm{r}, \sigma)]$

Scoring

Competitors will be ranked based on the value of their portfolio at the end of each round. The value of their options positions will be determined based on the intrinsic value of the position at the end of the round. The value of their stock positions will be determined based on the mid-market stock price at the end of the round.

Competitors ranks in each round will be squared and then summed to compute their final scores. The team with the lowest score wins. For scoring within the overall competition, the top placing team will receive 40 points, 2nd place will receive 39 points, etc.

Code Submission

We will require a preliminary submission by **noon (12:00 PM CST) on Friday, April 1st.**

Case Materials/Data

A development toolkit, detailed documentation and training data has been released through the UChicago Trading Competition Piazza. Training data consists solely of the price path of the underlying. We recommend checking your risk calculations against ours.

We have provided a bot that generates order flow to help you test. This is not how we will generate order flow in the competition and we will not be providing that order-generating bot to you. You can make a couple of assumptions about the competition day bot though: more orders will be sent to options that are more tightly quoted, more orders will be sent to options closer to being at the money, and orders will be sized proportionally to your Greek limits.

Questions

For questions regarding Case 2, please post in the UChicago Trading Competition Piazza in the “case2” folder.

CASE 3: Portfolio Risk Management and Asset Return Prediction

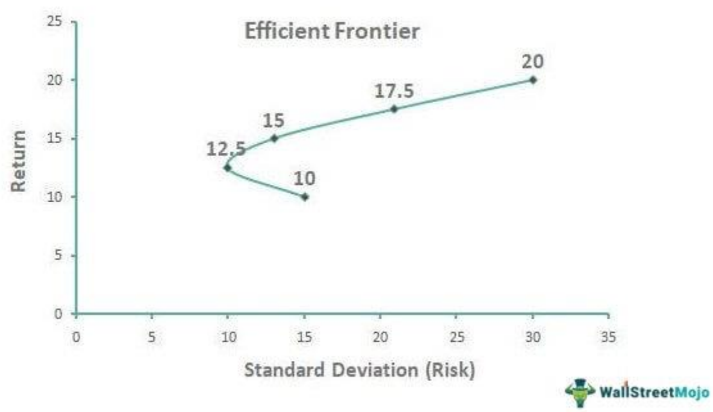
Introduction

In this case, you are a portfolio analyst tasked with allocating a fund over a ten-year time horizon across nine stocks your research team has selected. To aid you, your company has provided you with ten years of historical price data for each stock alongside a set of forecasts made each month one month ahead of time for each stock over that ten year period from three equity research analysts. Finally, you will be given the number of diluted shares outstanding for each company in the portfolio, which you may assume is always constant. Your goal is to develop an algorithm to construct and rebalance a portfolio aimed at maximizing returns while simultaneously minimizing returns variance.

Education

Markowitz

Obviously, a high expected return is a marker of success for any portfolio, but in addition to this, a great portfolio must also generate high returns consistently, i.e., investors must be certain their portfolio is not likely to lose money over significant periods of time. Therefore, portfolios should minimize risk while maximizing expected return. In 1952, Harry Markowitz of the University of Chicago published "Portfolio Selection," which formalized this basic intuition and formed the basis of modern portfolio theory (MPT). Markowitz conceived of an efficient frontier of portfolios that maximize return given a certain level of risk (or alternatively minimize risk given a certain desired return). Along this efficient frontier, the actual portfolio chosen is based on the individual investor's level of risk aversion (high risk aversion = low level of risk, and vice versa). In the absence of exact knowledge on risk aversion, portfolio managers can choose an efficient frontier portfolio based on the specific objectives given to them by their clients or firms.



Obviously, in order to implement a Markowitz portfolio, investors need some estimate of expected return and an estimate of risk. Additionally, we need some way to estimate the correlations, or covariances, between the returns of the different assets. Intuitively, a portfolio where all asset returns are highly correlated should have more risk than one with the same weights and risks for each individual asset but where all the asset returns are uncorrelated due to the fact that in the former case, one asset losing a large portion of its value means the entire portfolio likely will while this is not true in the latter case. Indeed, we see that the variance of returns in a portfolio (a measure of risk) is given by

$$\begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \begin{bmatrix} \mathrm{Var}(r_1) & \mathrm{Cov}(r_1, r_2) & \cdots & \mathrm{Cov}(r_1, r_n) \\ \mathrm{Cov}(r_2, r_1) & \mathrm{Var}(r_2) & \cdots & \mathrm{Cov}(r_2, r_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(r_n, r_1) & \mathrm{Cov}(r_n, r_2) & \cdots & \mathrm{Var}(r_n) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

where wi refers to the weight of the ith asset in the portfolio, ri refers to the return of that asset, and n refers to the number of assets in the portfolio. While we can reasonably estimate variances and covariances of asset returns based on historical data, empirically, historical returns have not been a strong predictor of future expected return, leading to practical difficulties in implementing a Markowitz portfolio. Since the development of MPT, investors and researchers have looked for new ways of dealing with this problem.

Risk Parity Allocation (RPA)
One approach to portfolio allocation is to simply ignore expected returns when determining how to allocate assets in a portfolio. The most naive way of doing this would be to simply find the portfolio with the lowest predicted risk, but this would simply be to invest in a risk free asset, and the return offered by such a portfolio would not entice many investors. What we want instead is a way to have a portfolio full of risky assets where more weight is given to those with lower risk. This can be done by ensuring that every asset's risk contribution to the portfolio is equal, meaning that less risky assets have more weight than riskier ones in order to equalize the risk contribution. Here, risk contribution is calculated as

$$\frac{w_i(\sum \mathbf{w})_i}{\sqrt{\mathbf{w}^T \sum \mathbf{w}}}$$

where w is the column vector of weights, and $\sum$ is the variance-covariance matrix from the previous equation. In this calculation, assets that are uncorrelated with the rest of the portfolio contribute less to risk than correlated assets with the same risk level, which is intuitive since a drop in the value of an uncorrelated asset does not contribute as much to overall losses for the portfolio as would a correlated asset. As it has been shown that asset volatilities and correlations are relatively stable over time, historical risk contribution can be used as a reliable estimate for risk contribution.

Although the theoretical basis for RPA has its foundations in the 1960s building on the work of Markowitz, the first official RPA portfolio was implemented in 1996. RPA portfolios consistently underperform Markowitz portfolios in terms of expected return, but these portfolios became popular in the wake of the 2008 financial crisis, when their values fell far less than Markowitz and other portfolio allocation schemes. It was believed following 2008 that this protection against risk was worth sacrificing returns, but following the stock market crash in 2020 due to the Covid-19 pandemic, risk parity funds fared just as poorly as others, and investors started questioning the utility of this form of portfolio allocation.

Black-Litterman
If we cannot use historical expected return to estimate future expected return, and we cannot ignore returns, we need a better method of estimating expected returns. In 1990, Fischer Black and Robert Litterman created the Black-Litterman model to do exactly this. In this model, implied expected returns and analyst predictions, generated through non-quantitative methods, are combined together to estimate expected returns for a model like Markowitz. In order to weigh analyst predictions against implied returns, predicted and historical covariance matrices of the returns of the different assets are used. We calculate implied expected returns $\prod$, as follows:

$$\Pi = \lambda \Sigma w_{mkt}$$

where λ is the risk aversion parameter (generally between 1 and 3) and wmkt is the column vector of market capitalization weights (what percentage of the market capitalization of all the assets in our portfolio belongs to a given asset). From here, we get the Black-Litterman formula for expected returns to use in our allocation scheme:

$$\left[\Sigma^{-1}+\Omega^{-1}\right]^{-1}\left[\Sigma^{-1}\Pi+\Omega^{-1}Q\right]$$

where Ω is the analyst generated covariance matrix of returns and Q is a column vector of analyst predicted expected returns.

When combining the Black-Litterman formula for expected returns with a Markowitz portfolio, investors can obtain higher returns than an RPA portfolio without having to worry about historical returns failing to predict future returns. However, in order to properly implement such a portfolio, it is necessary to generate both views on expected returns and views on the variance-covariance matrix of those returns, and there is no clear consensus on how to accomplish this. One final note is that although the Black-Litterman formula was designed to provide more accurate expected return calculations, there is also a Black-Litterman generated covariance matrix, which should be used when implementing Black-Litterman, given by

$$\left[\Sigma^{-1}+\Omega^{-1}\right]^{-1}$$

With this information in mind, successful competitors will, at a high level

1. Survey literature to understand portfolio allocation schemes and asset return prediction methods.
2. Construct an investment strategy based on data provided that utilizes accurate predictions of future prices while keeping risk low.

Case Specifications and Rules

The χ-Change trading platform will not be used for this case. Teams are expected to develop their strategies using our Python stub code and submit their code before the competition.

We will run each competitor's portfolio allocation algorithm on a test dataset with data generated by the same process as the data you are given; this test dataset will immediately follow the period in the training dataset. There will be one round, the results of which will be computed prior to the competition and played back during the competition as if unfolding in real time. As such, you must submit your final code to the case writers beforehand.

You may use any packages (and any programming language) to study the training data we will provide, but the submitted portfolio allocation code must be in Python and will be restricted in dependencies. The environment used to run submitted competitor code will be Python 3.10 and will only have the NumPy, pandas, and SciPy packages installed (alongside base Python). Although advanced and/or complex machine learning techniques are interesting to study and are valuable to learn, they are not the focus of this case and are not required for the purposes of solving this case. We strongly advise that you test your submission using a similar environment on your local machine before submitting your final code; submitted code that does not compile or that fails to run for any timestep will be disqualified for this case and the team that submitted it will receive 0 points.

In each timestep, asset prices and analyst predictions one month (21 days) ahead of the beginning of the current month (index is divisible by 21) will be provided and teams will submit portfolio allocations among the available assets for that period. These allocations will be in the form of weights on each stock: weights can be positive, negative, or zero in each timestep, and the submitted weight vector in each timestep will be L1 normalized before calculating portfolio returns. We assume there are no exchange fees or bid-ask spreads in the market and no liquidity concerns to deal with in allocating your portfolio.

Scoring

Times will be ranked based on their annual daily Sharpe ratio over the 10-year test period given by

$$S_p = \frac{\mathbb{E}[R_p]}{\sigma_{R_p}}$$

where RP is the daily return series of the portfolio. Normally, we calculate the Sharpe ratio based on the excess return series rather than just the return series, but for our purposes we will assume that the risk free rate means relatively constant and can be ignored. The team with the highest Sharpe ratio will receive 40 points, the next highest will receive 39 points, etc.

Case Materials/Data and Code Submission

Python stub code and training data will be released along with the case packet and additional supplementary resources through the UChicago Trading Competition Piazza.

We are requiring the final code for this case to be submitted by **noon (12:00 PM) CST on Friday, April 1st**. Note that this is different from Cases 1 and 2, as we will be computing the results of this round prior to the competition. Code submitted past this deadline will not be accepted, and we reserve the right to disqualify any competitors who submit incomplete code or miss this deadline. Again, **we strongly advise that you test your submission in a Python 3.8 environment with only NumPy, pandas, and SciPy installed before submitting your final code.**

Miscellaneous Tips

1. **Analyze returns, not prices**. Prices of stocks tend to be non-stationary processes, but returns are generally stationary. Analyzing returns series will be more fruitful for your strategies than analyzing price series.

2. **Don't test strategies on the same data you train them on**. Strategies will likely perform well on data your model has already seen - what's relevant is how well the strategy performs on data the model has not yet seen. You should not necessarily expect that your strategy will perform as well out-of-sample as it will in-sample; holding out a portion of your training data to test on (or running any other procedure to test on new data) is strongly advisable to get a more accurate sense of how successful your strategy will be.

3. **Pay attention to day count conventions**. Note that the Sharpe scoring formula implies that each year consists of 252 trading days, and thus each month consists of 21 trading days. In addition, the risk-free rate is reported in an annualized form; you will have to convert it to a daily rate using the day-count convention.

Questions

For questions regarding Case 3, please post in the UChicago Trading Competition Piazza in the "case3" folder.

Appendix: Liquidity and Exchange Primer

Liquidity as Bilateral Search

Search models are often referred to as matching problems while they are mathematically formulated as Optimal Search problems. We commonly refer to the two parties in bilateral search as active and passive, where one party (active traders) are seeking matches while the other (passive traders) wait for orders to come to them. In a search model, you no longer can interact with orders previously passed over and must choose to accept orders as they come.

Naturally, this is formulated for an active trader as a dynamic programming problem where we maximize our expected value through accepting a match and ending our search or proceeding with our search. Passive traders, meanwhile, post limit orders or quote on markets that may potentially lead to committing to prices that are posted. Note that spoofing, placing orders without the intention of committing to that price, is not permitted on our exchange.

Dimensions of Liquidity

We commonly refer to three dimensions of liquidity.

- Immediacy: How quickly trades of a given size can be arranged at a given cost. Traders generally use market orders to demand immediate trades.

- Width: The cost of doing a trade of a given size. For our purposes, we can identify width as the current bid/ask spread defined as best ask less the best bid. Width is the cost per unit of liquidity.

- Depth refers to the size of a trade that can be arranged at a given cost. Depth is measured in units available at a given price of liquidity.

Mathematicians say that breadth and depth are duals to each other. Minimizing the cost of trading a given size is identical to maximizing the size of a trade at a given cost.

Why Trade

Impatient traders focus primarily on immediacy and its cost. If we were to foreshadow price and discover profitable trades in our search, we may choose to accept these orders, potentially focusing on different aspects of the search problem. To summarize, liquidity is the ability to quickly trade large sizes at low cost. Traders are far more willing to trade with uninformed traders than with well-informed traders. Liquidity thus is different for traders who are known to be uninformed than for those who are known as informed traders. Markets may be liquid for the former but not for the latter. In practice, and especially in this case, traders often do not know who is well informed. Traders who can convince others that they are not well informed to obtain better prices or more size. Instruments that interest large numbers of traders and instruments with well-known fundamental prices tend to trade in more liquid markets. If other traders are buying more often or selling more often, it may leave imbalances in the order book. Imbalances will lead to liquidity being cheap for one side and expensive for the other side.

Origins of Asset Volatility

We distinguish between two concepts of volatility, fundamental volatility and transitory volatility. Fundamental volatility occurs when fundamental factors influencing values of assets change. Examples include unexpected changes in weather conditions in Case 1 or a surprising shift in volatility regimes in Case 2. Note that expected changes in these fundamental factors do not change prices as available information is encapsulated in informative prices. Transitory volatility exists when demands of uninformed and/or impatient traders cause prices to diverge from fundamental values. Examples include bid/ask bounce, the shifting of transaction prices within the range of the bid and ask due to trades on the bid or ask. Separating the two components is critical when reflecting on prior trades and market participation.

Measuring Transaction Costs

Many traders measure the liquidity cost as the absolute difference between trade execution price and quotation midpoint, the average of bid and ask prices. Depending on when we measure the quotation midpoint, the transaction cost differs. Measurements of the quotation midpoint at time of trade is called the effective spread while post-trade quotation midpoints produce realized spreads. Since prices shift in response to aggressive traders, where prices rise in response to aggressive buyers and prices fall due to aggressive sellers, realized spreads are typically less than effective spreads. Effective spread less realized spread is a measurement taking into account post-trade price movements, often used to measure market making losses to well-informed traders.

Summary

The discussion above concerns the nature of matching orders within a market and how to view liquidity from the perspectives of various participants in the market. Beyond the discussion of liquidity are a range of confounding factors that may affect the ability to provide "better liquidity" in the market through tighter quotes (Width) or larger volume (Depth). In particular, the notions of categorizing volatility and methods of measuring adverse selection introduce a possible mechanism for

Citation:

Harris, Larry. "LIQUIDITY DIMENSIONS." In Trading and Exchanges: Market Microstructure for Practitioners. New York: Oxford University Press, 2003.

χ-Change FIFO Allocation

As market participants look towards exchanges for liquidity, it necessitates a rule-based order-matching system. Our exchange allows competitors to negotiate prices through limit order quote placement, order cancellation, and a variety of aggressor order placements. FIFO allocation schemes are prevalent among nearly every major exchange.

FIFO, also known as the price/time algorithm, uses only two criteria for order matching - first price and second time. First, incoming market orders are filled at the best possible price standing within the order book. For any given price level, the orders are then filled in terms of time priority. In practice, FIFO encourages participants to not join existing order levels, but rather narrow spreads if they would prefer to interact within the market.

Let us consider the following example, where we have standing asks and an aggressor looks for a market order of 200 lots. Assume the order book resembles

Ask Size	Ask Price	Time
100	0.52	00:00:01.052

50	0.52	00:00:01.227
100	0.52	00:00:04.195
200	0.52	00:00:05.160
10	0.53	00:00:08:902

Then the order will fill at the level of $0.52 using the first three orders. Based on time priority, Asker 1 will obtain all 100 lots, Asker 2 will obtain 50 lots, and Asker 3 will be filled for 50 lots. Afterwards, the order book will look like

Ask Size	Ask Price	Time
50	0.52	00:00:04.195
200	0.52	00:00:05.160
10	0.53	00:00:08:902