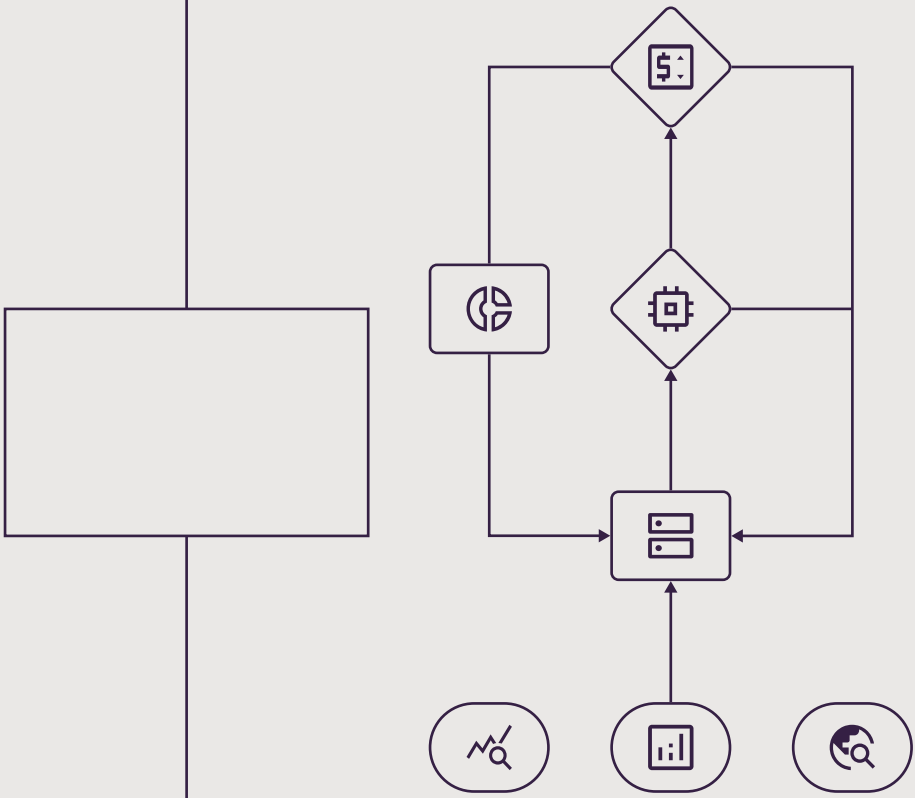


ALGORITHMIC TRADING THEORY AND PRACTICE

A PRACTICAL GUIDE WITH APPLICATIONS ON THE VIETNAMESE STOCK MARKET

ALGOTRADE
THE GIOI
THE GIOI PUBLISHERS

ALGORITHMIC TRADING THEORY AND PRACTICE
A Practical Guide With Applications on the Vietnamese Stock Market

Published by ALGOTRADE, LLC.

TABLE OF CONTENTS

CHAPTER III. DATA

CHAPTER IV. BACKTESTING

CHAPTER V. OPTIMIZATION

CHAPTER VI. FORWARD TESTING

CHAPTER VII. REAL TRADING

CHAPTER VIII. EVALUATION CRITERIA FOR TRADING ALGORITHMS

CHAPTER IX. OPTIMIZATION IN MULTI-ALGORITHM TRADING

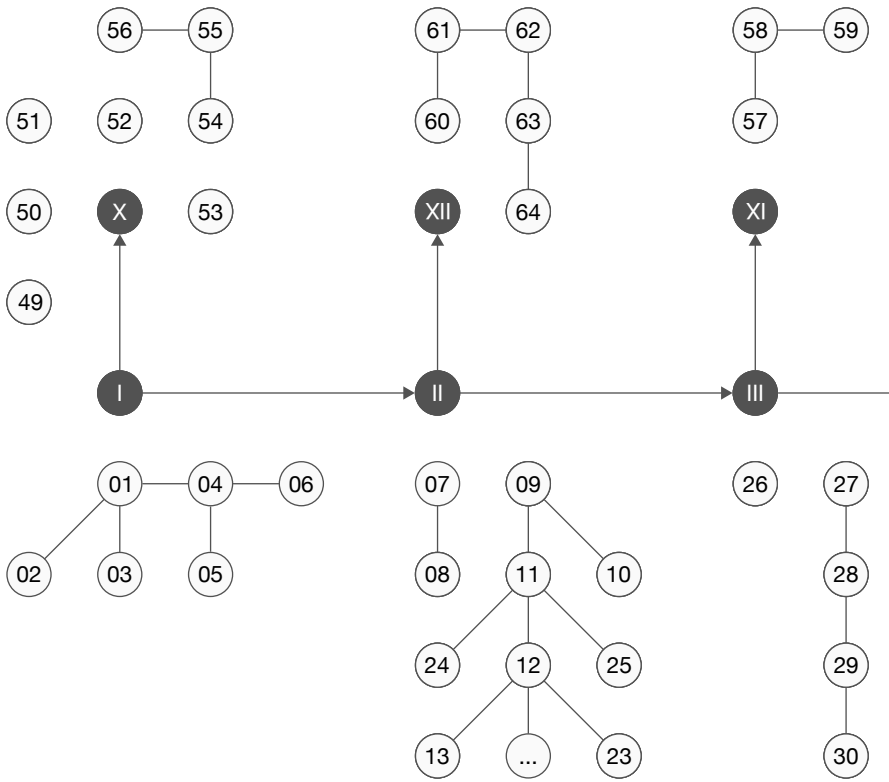
CHAPTER X. ALGORITHMIC TRADING PRACTICE

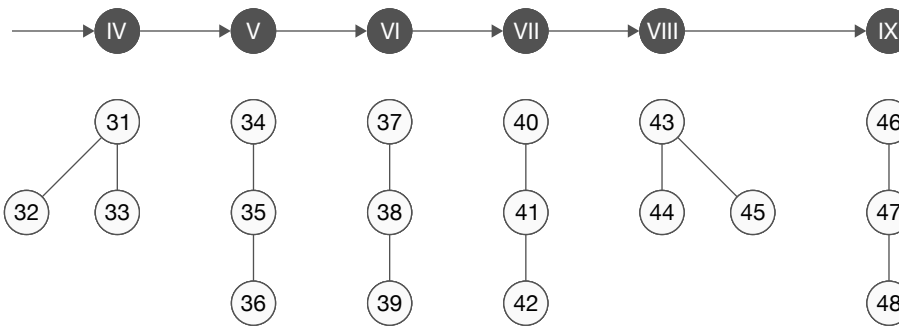
CHAPTER XI. INTEL CENTER – SUPPORT CHANNEL

CHAPTER XII. ALGOTRADE LAB – FIRST STEP TO ALGORITHMIC TRADING

GLOSSARY

READING DIAGRAM




IV
V
VI
VII
VIII
IX
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

PREFACE

You have in place profitable trading strategies in place and tested in the market for years but too time-consuming to execute. Do you like to completely automate this process to spend your time on more important work?

Do you have new trading ideas but are unable to assess their performance in the long term? Do you find potential strategies in testing but end up losing in real trading?

Knowing the limitations of AmiBroker, MetaTrader, or TradingView, do you want to develop your own trading system with the capability to deploy a diverse range of trading strategies, including but not limited to technical analysis? Do you want to approach a high-frequency trading system but face too many technical barriers?

Are you a fundamental analyst investor looking for technology to optimize the trade execution process in Vietnam?

Algorithmic Trading Theory and Practice: A Practical Guide With Applications on the Vietnamese Stock Market is a book for you. It covers the entire process of building an algorithmic trading system and a roadmap to turn ideas into real investment strategies, testing, optimizing, and automating the entire execution process. Here are the main contents:

Chapter I provides core concepts and basic algorithmic trading knowledge.

Chapter II to **Chapter VIII** provide detailed instructions on how to build an algorithm to trade on real markets, including data, APIs, trading strategies, backtesting, optimization, forward testing, live trading, and algorithmic evaluation criteria.

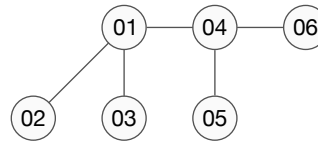
Chapter IX is dedicated to multi-algorithm trading.

Chapter X outlines practical experiences in algorithmic trading in Vietnam.

Chapter XI offers a free application that supports the decision-making process when trading.

Chapter XII allows readers to experience trading algorithms on personal investment accounts following a simple moving average strategy.

The ALGOTRADE team hopes you find helpful information in our book. We extremely appreciate your support.



CHAPTER I

ALGORITHMIC TRADING OVERVIEW

01

INTRODUCTION TO ALGORITHMIC TRADING

How long do you think one takes to evaluate this expression?

$$260 \times 252 - 721 = ?$$

No matter how fast one can calculate the expression, one cannot solve it faster, more reliably, and more accurately than a computer. This is the fundamental concept of algorithmic trading: making use of computing power to perform tasks in which computers outperform humans' capabilities.

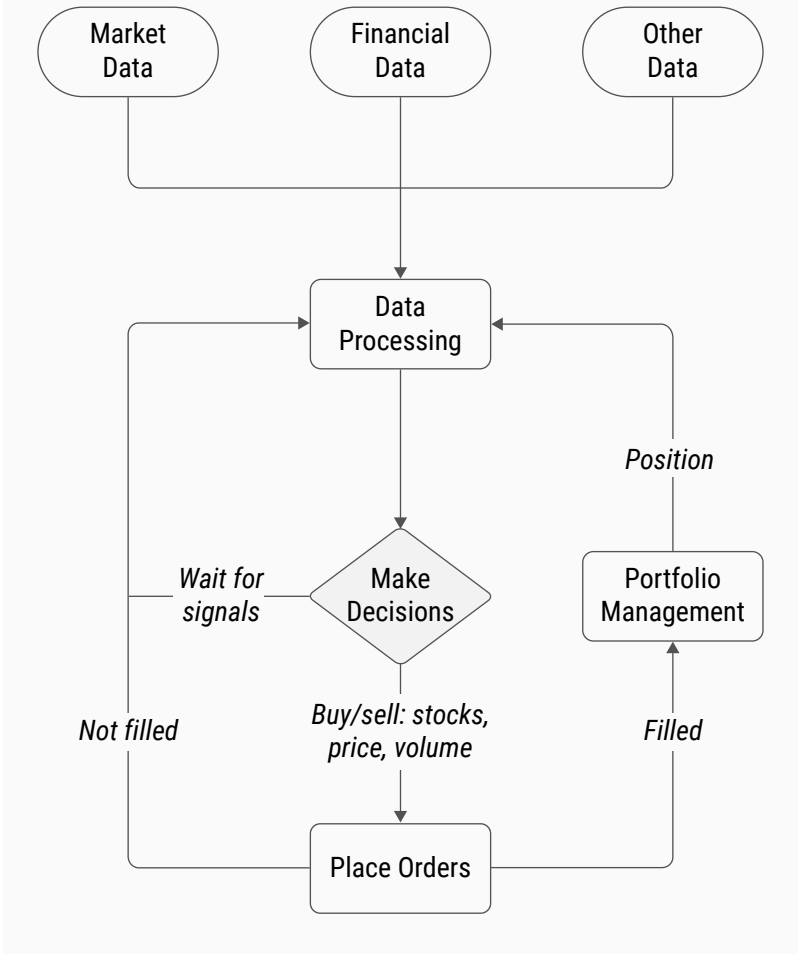
Algorithmic trading is using a computer system to carry out fully automated trades according to pre-programmed algorithms.

A **trading algorithm** is a concrete investment strategy in a set of execution statements to make trading decisions. It also determines what to buy or sell, at what price, and in how much volume and in which order type.

An **algorithmic trading system** automates all data collection processes, and data queries and uses computer algorithms to make trading decisions and reports, to manage real-time financial portfolios without any human intervention.

- **Query Data.** An algorithmic trading system can automatically update the latest information on the price and volume of all stocks in companies listed on all 03 stock exchanges in Vietnam, namely HOSE/HNX/UPCOM every 02 seconds, and it can query data at all times. This is a huge competitive advantage for investors who want to exhaust existing opportunities in the market.

Figure 01 Algorithmic Trading System



- **Make Trading Decisions.** The system automatically makes trading decisions after each data update. A high-frequency trading algorithm can make thousands of trading decisions every day.

- **Place and Cancel Orders.** After making a trading decision, the algorithm can place or cancel trading orders to the investor's account. It is possible to successfully place trading orders as fast as 50 milliseconds in the Vietnamese stock market.
- **Report Trading Results.** The system can report signals and results of placing/executing/canceling orders in real time. Simultaneously, it constantly updates the account status and summarizes trading results. In addition, it reports bugs in real-time to improve reaction time and resolve technical problems.
- **Manage Financial Portfolios.** After updating the portfolio, the change in the trading position will affect subsequent trading decisions. Portfolio management gets more complex as the trading system operates on both the stock market and the derivatives market.

If there's no delay in order matching, all the steps above can be completed in 2.5 seconds by a computer system, compared to manual operations which can take over 30 seconds.

Thanks to the algorithm process, the strategy can be consistently executed on a large number of transactions, eliminating human biases (emotional biases and cognitive errors), thus following the law of large numbers to assess the effectiveness of trading strategies.

It's important to note that the investment performance mainly depends on the algorithm design. A poor investment strategy will only lead to automatic losses when put into an algorithmic trading system.

02

ADVANTAGES OF ALGORITHMIC TRADING

Algorithmic trading brings unique strengths of computer systems to traders that traditional trading does not have. Taking advantage of the algorithmic potential can dramatically change investment performance.

The Law of Large Numbers

In probability and statistics, the law of the large numbers states that if the same experiment is repeated independently a large number of times, the average result must be close to the expected value. Alternatively, even with random and unknown events, a sufficiently large number of attempts will lead to a stable average result.

The law of large numbers can be illustrated by rolling the dice. The probability of the occurrence of all sides 1, 2, 3, 4, 5, and 6 is the same. Therefore the expected value of a dice roll event is:

$$(1 + 2 + 3 + 4 + 5 + 6) \div 6 = 3.5$$

If one only rolls the dice a few times, the average value of the results may be far off from this expected value. For example, roll the dice 3 times and the results are 6, 4, and 5 respectively. The average value is thus 5. However, according to the law of large numbers, if one rolls the dice many thousands of times, the average result will come closer to 3.5.

In trading, the winning or losing results of a handful of transactions can occur randomly and unpredictably. According to the law of large numbers, the results from a small number

of trading orders are not sufficient to judge its effectiveness. However, if there's a reasonable trading strategy with a winning expectation over 50% across a large number of trades, then on average it will generate profits in the long run even though there might be periods of consistent losses. Moreover, these results can be used as tests to evaluate a trading strategy's performance.

In practice, there's a big issue when applying the law of large numbers to real-life trading. It's often difficult for investors to ensure all transactions are sufficiently consistent according to a predefined strategy, due to the following factors:

- Effects of psychology and emotions;
- Risks such as mistakes in placing orders, or slow reaction time that leads to missing buy or sell opportunities;
- Unable to place orders according to the predefined strategy due to irregular price monitoring.

Because of these issues, the results obtained can be far from the initial expectation despite an effective trading strategy and a sufficiently large number of trades. More importantly, it's impossible to assess whether this result is due to a low-performing strategy or poor execution. An algorithmic trading system can solve these problems. It ensures the application of the law of large numbers, minimizing the impact of undesirable factors. Investors only need to focus on building, evaluating, adjusting, and improving their strategies to create stable long-term profits.

Ensure Investment Consistency and Eliminate Emotional Interference

In stock trading, besides a clear strategy, emotional management is a key factor of investment results. Investors are often swayed by emotions and make decisions contrary to their investment principles or strategies.

For example, before the 2008 global financial crisis, the market signaled that a crisis was imminent. Yet a lot of investors ignored these signs because they were caught up with the "frenzy" up-trend momentum in the market. Few thought a crisis was likely to happen.

Algorithmic trading solves this problem by ensuring all trades follow a predetermined set of rules and strategies. It eliminates the emotional influence of trading altogether. A simple example is the 7% stop-loss principle, common among Vietnamese investors to avoid big losses. However, very few investors actually carry out this trading principle. They often keep holding on to falling stocks even when the price has fallen too low. A trading algorithm will help investors avoid this. A stop-loss strategy does not guarantee good results, but it does ensure consistent investment principles.

Increase Confidence, and Reduce Stress and Insecurity When Investing

There are random unpredictable events every day. They are out of control yet have a direct impact on the market and stock prices. These events require investors to make a decision to buy, sell or keep holding. Making uninformed decisions gives most investors ambiguity, stress, anxiety, and insecurity.

The algorithmic trading system is not a magic wand that can tell the future and ensure all trading decisions are correct. However, it is an automated system that quickly reacts to random events that can happen without human intervention.

Investors only need to set up periodic reviews, monthly or quarterly, to evaluate an algorithm's performance. They can make necessary adjustments and upgrades to optimize profits. This mechanism boosts investors' confidence and reduces stress because there's no need to be hands-on every day.

Minimize Market Impact When Trading Large Volumes

A high-volume transaction from institutional investors, hedge funds, or major shareholders can dramatically change the market equilibrium. To avoid this, investors usually split their orders into several small-volume trades.

For example, if you would like to buy 5 million shares of FPT, investors can divide it into 1000 orders, each at 5000 shares per order every 5 minutes. After one hour of this execution, the investors assess the impact of the orders on the stock price. If the share price doesn't change much in the unfavorable direction, keep following the same strategy and check in on the price every hour until reaching the target of 5 million shares.

However, there's a time disadvantage to this strategy. In this example, even under favorable market conditions, it still requires at least 83 hours or 21 trading sessions to complete.

An algorithmic trading system solves this problem by simultaneously calculating the volume and making the stock purchase. It also checks the market impact immediately to significantly reduce the number of buy orders and the time required.

Save Time

A complete algorithmic trading will ensure full execution of the investment process without any investor intervention. It is the best benefit as investors can spend their time more efficiently instead of closely monitoring live boards all day long to place orders.

03

ALGORITHMIC TRADING RISKS

It's possible for algorithmic trading systems to make errors indeed. Although rare, mistakes can still occur at the most unexpected times. Some serious mishaps can wipe out the entire fund in a matter of seconds. It's required to have quality risk management for a sustainable system. There are 3 main risks that we found.

Buy and Sell Loop

This is the top risk that can occur in algorithmic trading systems. When a loop of buying high and selling low occurs, 99% of an account in terms of asset value can be lost in just a minute.

Knight Capital, a market maker, lost $440 million in just 45 minutes on August 1, 2012. A new trading algorithm at Knight mistakenly traded 150 stocks, buying higher and selling lower than the market price. Knight was on the brink of bankruptcy when the issue had been flagged. A loop error resulted in a disastrous loss to Knight just in 45 minutes.

Data Error

When data errors occur, like delayed data, the trading system may start to make arbitrary decisions. It is a state in which the system makes trades not based on real-time data. It leads to serious losses and inconsistent performance system-wide. At ALGOTRADE, we use at least 3 different data sources to cross-check in real time and to minimize data errors.

Unexpectedly Low Liquidity

Investors can suffer big losses if they only focus on the latest price and disregard liquidity with market orders. This may not impact small individual accounts; however, the effect is much larger for institutional accounts. In particular, this effect will have a huge impact on multi-algorithm trading with strategies using market orders to open and close positions in the same direction.

On May 6, 2010, the Flash Crash happened in the U.S. market. It was a stock sell-off that led to a sharp price fall in a matter of minutes. Without taking into account liquidity issues, trading orders were executed with a 60% price difference from before the Flash Crash event. It almost wiped out the entire account balances of the parties involved.

Serious risk management to avoid the instant collapse of an algorithmic trading system is extremely important and should be of the highest priority. Other risks related to computer programs such as API and connection errors may occur frequently. Yet these errors are much less likely to crash the entire system.

Iterative improvement is pivotal to enhance a system's stability. However, investors shouldn't stress too much over small, unavoidable bugs. Even though it may open an arbitrary position, it may not always lead to loss. In fact, erroneous trading positions may still unexpectedly make profits.

04

CORE COMPONENTS TO BUILD AN ALGORITHMIC TRADING SYSTEM

When building an algorithmic trading system in Vietnam, for technical analysis in the derivatives market, investors can use Entrade or third-party software such as AmiBroker and MetaTrader.

If investors need a complete system that can deploy a variety of algorithms including but not limited to fundamental analysis, multi-data (e.g. real-time oil price chart), or high-frequency trading systems, the following core components are critical.

- **Trading Algorithm.** Algorithms are the most crucial factor. If there's no stable profitable trading algorithm in the long term, a super-powerful system still cannot take advantage of computing power. In addition, shaping the trading strategy is a prerequisite to structuring the remaining system components. Specifically, a derivatives trading strategy works on vastly different data compared to a stock trading strategy.
- **Database.** Database stores price, volume, account, and transaction information with financial statements. Databases are often underinvested because they have little impact on algorithmic trading systems in real scenarios while they require lots of time and energy to maintain. However, for sustainable growth, algorithmic traders have no choice but to build high-performing databases. In the long run, databases

serve algorithm enhancement with research and serve as inputs to high-latency algorithms.

- **Stock Market APIs.** APIs allow placing and canceling orders as well as retrieving account information in real-time. There are many public security companies offering API services. This promotes the development of algorithmic trading in Vietnam in recent years.
- **Real-Time Trading Data.** The trading system relies on real-time trading data to make trading decisions. In addition, these data are also stored in databases for long-term use. In Vietnam, besides the APIs of security companies that come with data packages, investors can purchase data from providers such as FireAnt and Fialda. When building an automated trading system from scratch, investors normally use raw data from the security companies' API because of its high coverage, low latency, and free of charge. Data provider services are most suitable for MetaTrader or AmiBroker.

All algorithm prototypes are expressed via data and programming terms. Programming skills are thus required to build algorithmic trading systems. Python is a powerful and popular programming language for algorithmic trading. C is also a top choice for investors who need high execution speed.

05

SEMI-AUTOMATED TRADING

Semi-automated trading is a system that uses computers combined with expert input to optimize the decision-making process. Traders are primarily responsible for the decision-making process. They process information and data that computers haven't solved on their own, while computers bring speed, stability, and accuracy. There are many semi-automated trading systems to support investors. Let's take a look at examples in Vietnam's stock market from a simple to complex level.

Stock Filter – *Complexity: very simple.* It is a widely used application of semi-automated algorithmic trading. The trader defines a set of criteria and the system will find qualified stocks. This saves a lot of time by shortlisting the market-wide search to several potential stocks.

In Vietnam, there are misconceptions about equating stock filters and algorithmic trading. Stock filters only help to the degree of data query and give supporting information. In terms of complexity, a professional investor with strong Excel skills can build a good filter in 2 days. However, building a complete algorithmic trading system requires several years along with a solid understanding of database management, server administration, programming (Python, C), investment strategies, and risk and portfolio management.

Open/Close Position Schedule – *Complexity: simple*. The investors clearly understand the market situation and what actions to take if certain market conditions are met. Constantly checking market data to wait for conditions can be time-consuming. This leads to the use of a semi-automated trading system. Some common actions in this setup are price momentum trading, stop-loss, or take-profit.

Trading Recommendation – *Complexity: average*. While not common in Vietnam, this approach allows systems to suggest trading ideas. It requires an investor's approval to execute orders. It always leaves the final decision up to the traders.

Parameter Configuration – *Complexity: complex*. It is an algorithmic trading system where investors can interact with the system in real-time by changing algorithmic parameters. They can thus impact trading performance. The system may be fully automated using a default configuration. This setup is highly effective when expert experience is extremely valuable, or when the system is in risky situations that require human intervention.

Multi-Asset Class Trading – *Complexity: very complex*. Even when investors have great trading ideas, like pair trading for example, they will have difficulties realizing the ideas without the support of an automated trading system. Thus, a predefined set of rules that allows parameter configuration will be extremely useful for investors who want to implement a pair trading strategy.

06

09 STEPS TO DEVELOP ALGORITHMS

Building a fully automated trading system that is profitable and stable is the first challenge in this field. There are 9 steps to develop a trading algorithm as follows.

1. Forming Algorithmic Hypotheses

Traders begin to develop an algorithm according to familiar investment strategies or market assumptions that will likely bring profits. This step is to break down a trading strategy into a set of statements that execute trading logic to make buy or sell orders. These orders have key information like the order type, price, and volume.

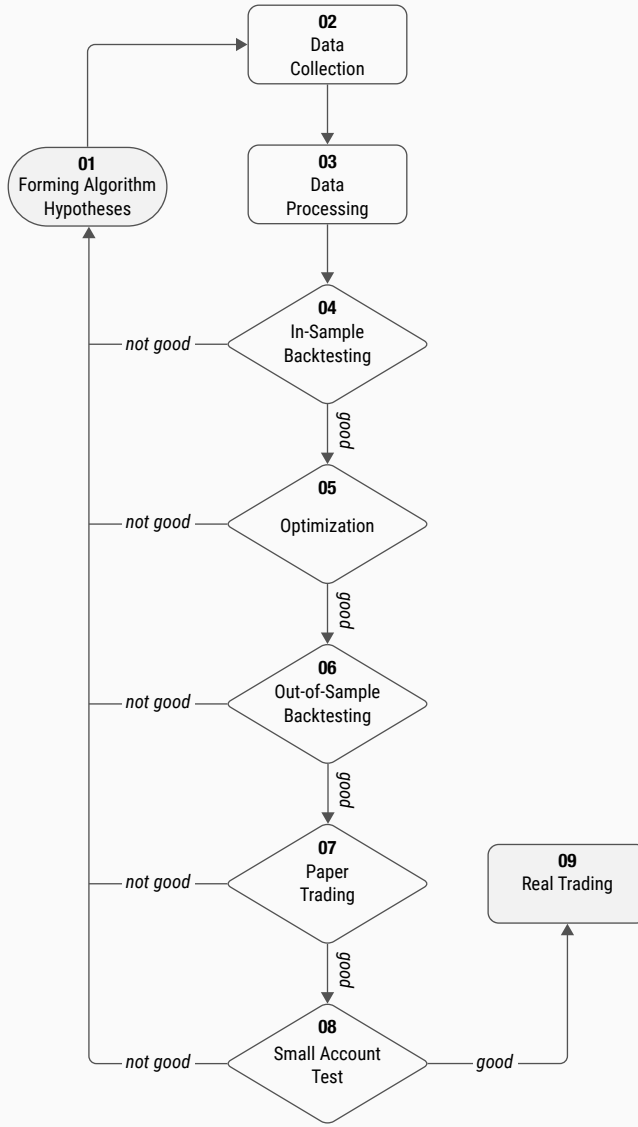
After suggesting the trading algorithm, it's possible to move on to real trading. However, it would be extremely risky and can be a game of chance. To limit the risks, testing phases are required especially with large accounts.

2. Data Collection

In most cases, basic data would satisfy the needs of algorithm development. However, for complex algorithms that may require tick data, financial statements, macroeconomic indices, commodities, or ETF indicators, traders need to collect data before implementing the algorithm.

Around the world, data collection has gone so far as using satellite images to estimate cars in a parking lot, or to estimate the number of sheep on a farm to predict business outcomes. This goes hand in hand with algorithm development to seek profits.

Figure 02 Algorithmic Development Process



3. Data Processing

This stage requires lots of effort to ensure that collected data is not duplicated. It also removes data outliers and structural errors to ensure sufficient quality data to develop algorithms.

Suppose investors in Vietnam would like to calculate the correlation between PVD shares and crude oil prices. After collecting oil price data, it is necessary to synchronize the time of world oil data with Vietnam time, adjusted for public holidays and any time zone difference. This is a simple example of the data processing operation.

4. Backtesting With In-Sample Data

Backtesting evaluates whether the trading algorithm is profitable. A correct method of testing on the historical in-sample data will give a reasonable expectation of its performance. It also helps understand the algorithm risk in the future.

To test past data, the first step is to divide this dataset into two parts, 70% and 30% of the dataset. The former is called in-sample data and the latter is out-of-sample historical data.

The in-sample data is used to evaluate the trading profile of the algorithm. The core metrics include profitability, maximum drawdown, and Sharpe ratio. Note that an algorithm requires at least 300 transactions to ensure these metrics are statistically significant. Testing on past data should also include taxes, fees, slippage, and any other trading costs incurred.

After estimating the parameter, if the performance is unsatisfactory, return to step 1 to propose a different hypothesis. Otherwise, the process can move to the next step.

5. Optimization

In this stage, investors make necessary adjustments to algorithm parameters to find the values that optimize the algorithm's efficiency in the targeted market. It is necessary to test hundreds

of different sets of parameters until a stable area is found. It is important to avoid overfitting, that is the set of parameters only yields good returns in the past but will unlikely be the same in real trading.

In Vietnam, many focus on finding a set of parameters with optimal profit results in past data to attract and sell trading algorithms to customers. Investors who buy and trade according to these algorithms have no idea they are in a highly risky game.

6. Backtesting on out-of-sample Past Data

After the optimization phase, traders continue to test the remaining 30% out-of-sample data from step 4. This process tests whether the post-optimization algorithm can generate stable profits. If the performance is similar to that of step 5, it's likely a good algorithm with strong potentials. Otherwise, investors should be cautious if they want to proceed.

7. Paper Trading

This stage tests the algorithm on completely unknown market data. The data comes from real data at the present until a certain point in time in the future. It is to objectively evaluate the algorithm's performance. If it is similar to that of step 6, investors can move to the next stage of testing on small accounts.

8. Small Account Test

After the 7 stages above, it's now possible to deploy the algorithm on a small account to test the performance, as well as the algorithm hypothesis during real trading. This stage also finds possible technical bugs and compares the result with the paper trading stage to make any adjustments if necessary.

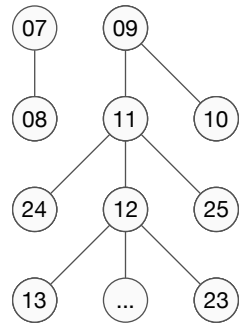
This period usually lasts 2 months to have a large enough number of transactions and a long enough time to assess the stability of the algorithmic trading system. If the performance

profile is still consistent with that of previous stages, investors now have confidence and scientific grounds to apply to real trades.

9. Real Trading

At this point, investors can confidently let the algorithm trade at maximum capacity. In the early stages, it's necessary to monitor the operation and fix any technical bugs that may arise. After 3 months, investors can officially launch a new trading algorithm if the operation proves stable.

Don't forget to get a cool and catchy name for your brand-new trading algorithm!



CHAPTER II

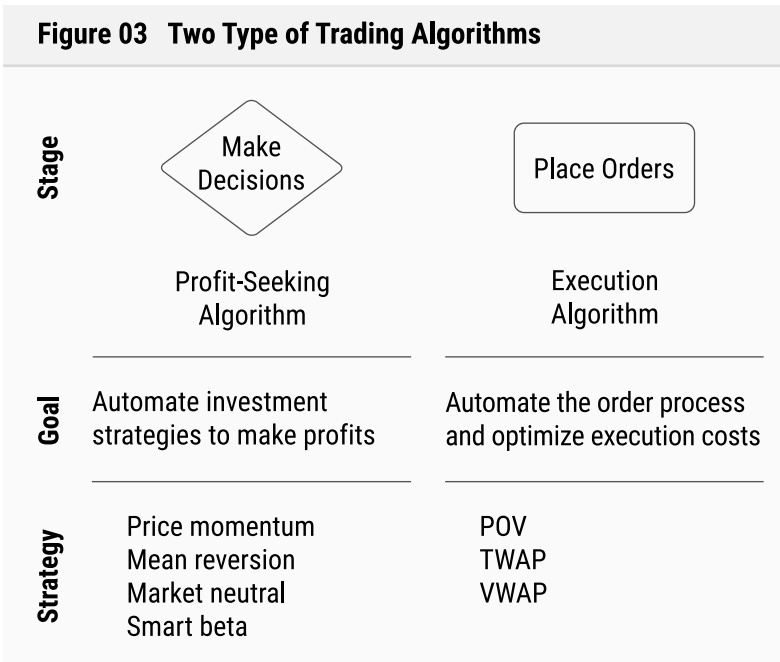
FORMING ALGORITHM HYPOTHESES

07

DISTINCTION BETWEEN TWO TYPES OF ALGORITHMS

An investment process always consists of two stages: making buy/sell decisions and executing these orders. Depending on applications, we can classify algorithms into two types: profit-seeking algorithms and execution algorithms.

Figure 03 Two Type of Trading Algorithms



Execution Algorithms

As the name suggests, these algorithms are used in execution orders to take advantage of computer systems. It is used to optimize the trading price. These algorithms are for high-volume trading and aim to avoid market fluctuations. Here are the common groups of execution algorithms:

- Scheduled algorithm: POV, TWAP, VWAP;
- Liquidity seeking algorithm;
- Arrival price algorithm;
- Dark strategy/liquidity aggregators;
- Smart order router algorithm.

Profit-Seeking Algorithms

Over the development of financial markets, profit-seeking algorithms have become more and more diverse. These algorithms all have the same goals to determine which securities to buy or sell, when, and in what volume to maximize profits.

The rules of this group are unified throughout the process from decision-making to order execution. The computer system will automate this process without human intervention.

The ultimate goal of investing is to increase profits. Investors pay more attention to the profit-seeking algorithms than the execution algorithms. However, it's not easy to find profitable algorithms among thousands of options and a complex changing market landscape. Therefore, in reality profit-seeking algorithms are still on a small scale though they get much attention. In contrast, the group of execution algorithms is less well-known but easier to implement and more stable. In fact, this group is often used on a large scale by investment funds and professional investors.

Note that in the following, referring to "*trading algorithms*" or "*algorithms*" without further explanation can be understood as referring to profit-seeking algorithms.

08

EXECUTION ALGORITHMS TO OPTIMIZE TRADING FEES

The execution price of a stock is determined daily by the rules of supply and demand. At any given time, if an investor executes an order to buy or sell shares in too large of a volume compared to normal liquidity, it will immediately create an imbalance of supply and demand. The stock price in the short term will move in an unfavorable direction to the investor. Specifically, when placing a high-volume buy order, the market price will increase, resulting in a higher average execution price than expected; and vice versa when selling in a large volume.

The larger the volume compared to the stock liquidity, the larger the impact on the market price in the unfavorable direction. For example, an investor places buy orders for stock A and stock B, at 1 million shares. However, the daily liquidity of these two is different (table below):

	Stock A	Stock B
Buy volume	1,000,000	1,000,000
Average liquidity from the last 10 sessions	50,000,000	2,000,000
Buy volume compared to average liquidity	2%	50%

While the volumes are the same, stock A has higher liquidity. So the buying order from stock A has less impact on the market price. The order to buy stock B is likely to have a big impact and increase its market price.

Why Large Volume Impacts the Market

When a high-volume buy and sell order enters the market, market participants are likely to believe that the market has key information that they are not aware of. They will then revise the price they are willing to buy or sell to reflect this unknown information.

To limit the market impact in an unfavorable direction, when trading in large volumes, investment funds and investors execute orders by dividing large orders into several smaller ones. They also execute orders at various times to minimize market impact. Commonly used execution algorithms include:

- POV algorithm (percentage of volume);
- VWAP algorithm (volume-weighted average price);
- TWAP algorithm (time-weighted average price).

For retail investors, these strategies may be of limited value. However, for large investment funds and investors, the application of execution algorithms is of great significance. It's no less than the efforts to find value stocks or to manage portfolio risks.

POV Algorithm

This algorithm executes orders at a predetermined percentage of market liquidity until reaching the target volume. As the trading volume in the market increases, the algorithm will trade more stocks and vice versa.

For example, investors like to buy 5 million FPT shares with the POV algorithm at a rate of 10%. For every 10,000 FPT shares traded on the market, the algorithm will buy 1,000 shares until reaching 5 million.

The advantage of this algorithm is that it keeps the volume of buy orders in the market depending on the current liquidity of the market (around 10% as in the example above). Thus it minimizes the impact on market prices.

The main disadvantage of this algorithm is that it's possible investors will not complete the trade within a specified period of time, especially in unfavorable market liquidity. In the example above, the historical liquidity of FPT at the time of writing is 2.5 million shares a day. It would then take 20 days to buy all 5 million shares if the liquidity stays constant.

However, at unfavorable times, market liquidity can drop to half only. It may take up to 40 days for the algorithm to complete. This leads to the risk that the fair market price has increased significantly from the expectation.

VWAP and TWAP algorithms can solve the problem above. They both calculate to divide the stock volume into small orders to trade and place these orders on a predetermined schedule. They ensure the algorithm is completed within a specified period of time, such as buying 5 million FPT shares completed in 20 days.

VWAP Algorithm

This algorithm breaks down and puts orders into the market in different time frames. The volume is calculated based on historical data. The challenge is that the historical liquidity data may not always repeat itself in the future. There are time frames with high trading volume in the past but this may not happen again in the future.

TWAP Algorithm

This algorithm breaks down and puts orders into equal parts and places them into the market at regular time periods (normally spaced 5 minutes apart). It should be noted that investors should split orders small enough to ignore market liquidity impact. It's because this algorithm does not include the liquidity factor in the calculation. Its goal is to complete the transaction within the specified time period.

In Vietnam, investors can see this algorithm is widely used on blue chip stocks. They are traded with a volume of fewer than 1,000 shares every 2 seconds.

09

06 COMPONENTS TO BUILD A COMPLETE TRADING ALGORITHM

In addition to two main components of position entry points and take-profit points, a trading algorithm also decides stop-loss points, target markets, position sizing, and trading tactics.

1. Position Entry Point
The entry point is the minimum requirement of any trading algorithm. Note that having an entry point does not mean the order is successfully executed. In many trading strategies, the fill rate can be under 10% in real trading.

2. Take-Profit Point
After the entry point, the timing or condition to take profits and rebalance portfolios is the next required condition for all algorithms. In certain special cases, the algorithm can take profits according to the time function, not the profit function. The profit would be more random in these cases.

3. Stop-Loss Point
Most algorithmic trading strategies should have stop-loss points while a few do not require them. In reality, it's likely hard to accept losses as a prevailing sentiment is "not cutting losses is not losing". However, with this mindset, investors can fall into the trap of winning lots of small pots but ultimately losing money in the entire process. This phenomenon is common in the mean-reversion and scalping strategies.

4. Target Market – Which Stocks to Buy/Sell

Experienced traders mostly trade in their familiar sectors. Every security has its own characteristics, formed by the impact of stakeholders. Since it attracts different groups of investors, funds, and traders, we never assume different securities work in similar ways. Any algorithmic trading strategy can follow this philosophy instead of a one-size-fits-all approach.

5. Position Sizing – How Much to Buy/Sell

In Vietnam, 95% of algorithm execution does not consider position sizing but buys or sells the entire account as soon as a position is opened. This may be due to small account size, or technical limitations of third-party software that do not allow position sizing to be adjusted. However, to allow multi-algorithm trading on the same account, it's necessary to take position sizing into account. Otherwise, buying or selling the entire account likely leads to systematic collapse within 03 years.

6. Trading Tactics – How to Buy/Sell

It's an enhanced version of using market orders in all situations to limit slippage costs. They greatly affect system-wide performance, especially for large accounts. Instead of using market orders, algorithms such as VWAP, TWAP, or POV can be used. In our system, ALGOTRADE uses an enhanced POV algorithm that allows high-frequency trading when there is a price advantage.

10

DIFFERENCES BETWEEN EQUITY AND DERIVATIVES MARKET

The number of investors using algorithmic trading systems in Vietnam is still limited. As an estimate, 98% of algorithmic trading systems in Vietnam operate on the derivatives market (VN30F futures contracts) but not on the equity market. This is due to fundamental differences between the two markets as follows.

Time to Complete Transactions

Differences in completion time explain why more than 60% of algorithmic trading prefers derivatives markets to stock markets. This can be understood as the algorithmic trading system that can trade at high frequency in the derivatives market. It's rather limited in the stock market since the wait time takes up to 1.5 days to sell.

Database Complexity

Most derivatives trading algorithms use technical analysis that includes only price and volume data. They are easy to collect and organize into a database.

In contrast, trading algorithms on the equity market use fundamental analysis, which requires full financial statements from all listed companies in Vietnam. They include income

statements, balance sheets, and cash flow statements. These reports usually have different structures when it comes to company types, like insurance, security companies, conventional companies, or even banks. Databases in the equity market have specific requirements in the data structure, query speed, data cleaning, etc. The database complexity is much higher in the equity market compared to the derivatives market.

Note that when using multivariate algorithms, the database gets even more complex. These algorithms require more types of data besides financial statements.

Fees and Taxes

Taxes and two-way transaction fees in the derivatives market are only about 30% of the stock market, excluding price slippage. From this perspective, the derivatives market looks more attractive to investors. However, in the long term, trading in the derivatives market may incur higher fees if investors opt for high-frequency trades. There's no daily transaction limit in the derivatives market.

In summary, there are three main differences between algorithmic trading in the equity market and in the derivatives market as follows:

	Equity Market	Derivatives Market
Completion time	T + 1.5	T + 0
Database complexity	High complexity	Low complexity
Fees and taxes*	0.35%	0.12%

* Estimated from security company fee tables.

For most algorithmic traders, the derivatives market is more attractive due to the three main differences above. However, at Algotrade, we are working on algorithmic trading for both markets, as fundamental analysis proves to be highly valuable. There are also opportunities for market-neutral strategies that combine both markets at the same time for higher efficiency.

11

HOW TO CONSTRUCT AN ALGORITHM HYPOTHESIS

Without a proper understanding of algorithmic hypotheses, it can be costly to waste time in subsequent stages without meaningful return. Finding a good algorithm may take a few years, even though forming one only takes minutes. A beginner in algorithmic trading will fail 99.5% of the time. Increasing the success rate from 0.5% to 10% is already a great success. This section demonstrates how to improve the success rate of an algorithm hypothesis.

Master Foundation Knowledge

- **Data.** Input data is a prerequisite to any algorithm. It plays an important role in forming an algorithm hypothesis. For instance, limiting input data to price and volume will force algorithmic traders to have technical analysis as the only choice.
- **Strategy.** The investment strategy must be concretized into a set of principles that must be followed when making trading decisions. It decides what securities to buy or sell, what type of order, when and what price, and in how much volume.
  For any investment strategy, the overall goal is to generate the highest possible return with the lowest risk. Understanding the risk-return trade-offs will enhance your chance of building a successful trading algorithm. Don't focus too much on a single strategy. In fact, the majority of algorithmic traders

only focus on the price momentum strategy because it is easy to program and execute. Imagine what would happen if 99% of traders share the same strategy?

Multiple strategies will limit risks in the long term. Articles 12 to 23 detail common trading strategies. Algorithmic traders can refer to these strategies to develop their own trading algorithms. They can be tailored to individual tendencies to become more effective.

- **Individual Tendencies.** Every trader has their own personality tendencies. Some prefer safety to 'high risk, high return'. Some are willing to bet their entire portfolio on a few positions, instead of dividing them into smaller orders to lower risks. Some focus only on high-frequency trading strategies, expecting to find profits in the short term while ignoring long-term strategies.

  Algorithmic traders may suffer from emotional discomfort when downplaying their individual tastes when forming algorithm hypotheses. It's because trading algorithms can operate wildly differently from their own personality. Understanding emotional tendencies will make the journey much simpler. For example, an algorithmic trader with low-risk tolerance who follows a price momentum strategy will find it difficult to stay calm with temporary losses. They often elect to intervene at unnecessary moments.

Real Market Experience

Proposing a good algorithmic hypothesis often comes from an already effective approach in discretionary trading. Personal trading experience with both wins and losses is always valuable for algorithmic hypotheses. This source of information is irreplaceable since it comes from real experience. Knowledge

from books and training courses is helpful but still cannot replace real trading. Thus algorithmic traders should take experience from decision-based trading, especially in the beginning. Ten years of market ups and downs will form a solid foundation for any investor.

Learn Proven Successful Trading Strategies

There are countless publicly available trading examples and strategies for free. Compared to a brand new, unproven high-tech strategy, investors can increase success rate based on proven trading strategies. Here are some examples:

- Warren Buffett's investment philosophy on value investing on undervalued stocks.
- Passive investment in SPY.
- International arbitrage business.
- Market makers in commodities and forex markets.

Examining the potential of an algorithm hypothesis before the start of the testing phase will save a lot of time and effort. It also increases the success likelihood at the next stage of algorithm development.

12

TRADING STRATEGY OVERVIEW

Trading strategies are one of the core components of an algorithmic trading system. Without stable, profitable strategies in the long term, an automated system cannot take advantage of the computing power. Even though different strategies may work in different markets, there are major groups of investment strategies below.

Market Neutral

Market neutral is a group of investment strategies undertaken by an investor who simultaneously opens long and short positions in order to minimize market risk on portfolio returns.

Pair trading is a simple and common form of market-neutral strategy.

Statistical arbitrage is also a group of market-neutral strategies that evolved from pair trading. It uses mathematical and statistical models along with computer assistance to make the most of trading opportunities. They may come from abnormal changes in the relative price of one stock compared to another. Statistical arbitrage is difficult for retail investors to access, and ALGOTRADE is making continuous efforts to close the technology gap for these investors.

Price Momentum

Investors look to buy rising stocks or short-sell falling stocks. This strategy believes that stocks continue to follow the momentum in the short term.

During the Covid years, this was the most widely used strategy in Vietnam. There's no guarantee, however, this approach will continue to be effective in the long term.

Mean Reversion

This strategy assumes that if a stock's price is too low compared to its intrinsic value, investors should buy and vice versa. In some cases, the intrinsic value may be replaced by the technical mean.

Event-Driven

Event-driven trading strategies take advantage of market inefficiencies from corporate events like mergers and acquisitions, corporate restructures, share buybacks, extraordinary dividends, etc. to trade in the short term. Investors study the situations surrounding these events and assess how they will impact the stock price. They identify potential opportunities that may arise and take advantage of them.

Market-Making

This strategy commonly places orders simultaneously at the best bid price and the best ask price. It makes profits from the bid-ask spread.

Scalping

It is a special strategy that focuses on an ultra-short time frame to open and close positions in order to make very small profits. Scalping traders make a lot of transactions daily and expect a very high success rate.

Front-Running ETF

Investors anticipate the action of exchange-traded funds (ETFs) according to the public prospectus and simulate part of the action prior to the fund's rebalancing.

Arbitrage

Arbitrage is a strategy that takes advantage of a temporary difference in the price of the same asset in two different markets. This strategy trades to make a profit without incurring too much risk.

Grid

This strategy sets up a price grid around a predefined value to make profits from market fluctuations. It works best in an oscillating market with large swings, though having a big risk in trending markets.

Smart-Beta

Known as a factor-based strategy, this strategy builds portfolios according to rule-based processes. It uses business factors like liquidity, value, and quality as criteria for making trading decisions. It is not widely used yet because it's necessary to establish an automatic connection to the financial data of all companies in the market.

Sniffing

This strategy relies on real-time transaction data to track the actions of individuals or organizations with large trading volumes. It assumes that these actions will last for days or weeks, and information advantage will ultimately generate significant profits.

13

MARKET-NEUTRAL STRATEGY

Definition

Market neutral is a group of investment strategies undertaken by an investor who simultaneously opens long and short positions in order to minimize market risk on portfolio returns. It's useful when investors want to remove market volatility from portfolio returns to focus on forecasting stock returns. This strategy has two main characteristics:

- Can yield profits regardless of whether the market rises or falls;
- Open opposite directions simultaneously with equal market risk.

In the stock market, investors often use beta from the Capital Asset Pricing Model (CAPM) as a measure of systemic risk. In this context, a market-neutral portfolio is defined as a portfolio with a beta of 0.

Pairs trading is a simple and common form of market neutral. The purpose of a pair trading strategy is to look for stock pairs that have a highly correlated price history. When the price correlation of two stocks deviates from the long-term average, investors buy stocks that are underperforming while selling stocks that are outperforming. They expect that this deviation is temporary. When the price correlation of two stocks converges to the expected average, investors close positions to realize profits.

Statistical Arbitrage (StatArb) is also a group of market-neutral strategies that evolved from the pair trading strategy. It uses mathematical and statistical models along with computer assistance to make the most of the trading opportunities. These opportunities come from abnormal changes in the relative price of one stock to another.

In terms of classification, the statistical arbitrage strategy group includes pair trading. Pair trading can be considered a statistical arbitrage strategy. However, statistical arbitrage is not exactly pair trading. With the help of computers, statistical arbitrage strategies can combine buying/selling two portfolios with hundreds of different stocks. At that time, investors will select potential stocks and rank them according to relative valuation criteria. Then they buy "relatively cheap" stocks while selling "relatively expensive" stocks.

Examples of Market-Neutral Strategies

Suppose the investor has just analyzed and selected a portfolio of 5 stocks in VN30 as follows: FPT, HPG, TCB, VIC, VPB (referred to as "VN05"). Investors predict in the future, the VN05 portfolio will be better than the general market. As a benchmark, the general market is the VN30 index. They decide to invest 600 million VND according to the market-neutral strategy as follows:

- Buy 500 million VND in VN05;
- Deposit 100 million VND to open 5 short positions simultaneously (assume VN30F1M index is 1,000; margin is 20%; and VN30F1M price changes entirely correlated with VN30 index score).

Assume the VN05 portfolio has better performance than the general market. If the market increases, VN05 increases more.

And conversely, if the market falls, VN05 falls less. Excluding fees and taxes, assume 2 market scenarios with investment results as follows:

	Bull Market VN30: +10% VN05: +12%	**Bear Market** VN30: -10% VN05: -8%
Buy VN05	500,000,000 × (12%) = +60,000,000	500,000,000 × (-8%) = -40,000,000
Sell VN30F1M	1,000 x (-10%) x 100,000 x 5 = -50,000,000	1,000 x (10%) x 100,000 x 5 = +50,000,000
Total Profit/Loss	**+10,000,000**	**+10,000,000**

If the market rises, VN30 increases. The short position on VN30F1M incurs a loss. However, investors expect the VN05 portfolio to increase stronger than VN30 (12% vs 10%). The profit from the VN05 portfolio offset the entire loss from the short position. The final profit is 10 million VND.

Conversely, if the market falls, the VN05 portfolio incurs losses. However, the investors expect VN05 to fall less than the market (say -8% versus -10%). At the time, the profit from the short position offsets the entire loss from the VN05 portfolio. The final profit is also 10 million VND.

Strengths of the Market-Neutral Strategy

As in the example above, with the market-neutral strategy, investment results depend primarily on selecting the right stock

portfolio with better potential than the market average. It's not dependent on accurately forecasting short-term price movements. Using this strategy, investors only need to focus on researching, understanding, and choosing a good portfolio of stocks with solid fundamentals. The portfolio should have high-profit potential in the long term, not worrying about short-term market fluctuations. Short-term events appear very random, and difficult to predict properly. They can only be perceived when they have occurred, and there's no time to act at such moment. In addition, it saves a lot of time and effort from monitoring the market and daily fluctuations.

14

PRICE MOMENTUM STRATEGY

What Is Momentum?

In the stock market, stock prices always change due to the impact of supply and demand. When demand is greater than supply, the price starts to increase, and vice versa. There are several reasons why supply and demand are always changing. Common ones include positive macro impacts, good business news, and cash flows from new investors. These fundamentals give the stock incentive to rise in the beginning when demand outweighs supply.

Investment analysts observe that once a stock starts to rise, the bullish momentum will continue to stay the same for a certain period of time before stopping or turning bearish. This process is similar to throwing a coin vertically into the air. When the initial force is stronger than the earth's gravity, the coin begins to accelerate and move upwards in the first few seconds. With momentum, the coin continues for another period with decreased speed before falling down. This analogy leads to a common term in physics "momentum" to be applied to finance.

Price momentum in finance is the rate of increase/decrease in a security's price over a period of time. It's also known as the rate of price change and is used to measure the strength of a trend.

In algorithmic trading, the **momentum strategy** buys rising stocks and short-sells falling stocks. It argues that stocks tend to keep following the trend in the short term. This argument comes from the belief that once the trend is established and there is

sufficient momentum, it will continue to last a brief period of time for scalping to take profits.

In short, the main purpose is to buy stocks when the price has risen and sell when the price rises even more. Buy high and sell higher. The opposite of this strategy is the mean-reversion strategy. It buys stocks that are too low compared to their intrinsic value, or average value, and vice versa. Buy low and sell high. This strategy will be discussed in the next article.

Causes of the Momentum Effect

There are two plausible explanations for this effect. The first comes from studies of behavioral finance related to investor biases. Investors often overreact to information and events, leading to an increase in strength and prolonging trends.

The second is that investors receive and process information at different reaction times. When there's a positive event, early investors realize the potential of the stock and start buying. It initiates the price trend to start moving upward. Some other investors slowly realize the opportunity but still have high expectations and decide to buy at a higher price. They maintain the uptrend. Both have positive expectations but different action timing, resulting in an uptrend that may last for a brief period of time before stopping and reversing.

Investment Risks in the Price Momentum Strategy

This trading strategy works on a basic assumption: when a trend appears, it will last for a sufficient period of time to take profits from corresponding positions. However, in practice, there's no guarantee this assumption always holds true.

Daily events and news directly affect investor psychology and expectations. They create changes in supply and demand,

initiating trends. However, at the same time, investor sentiment also changes unpredictably during this brief period. A newly started uptrend can be short-lived when later macroeconomic news negatively impacts investors' sentiment in the whole market.

15

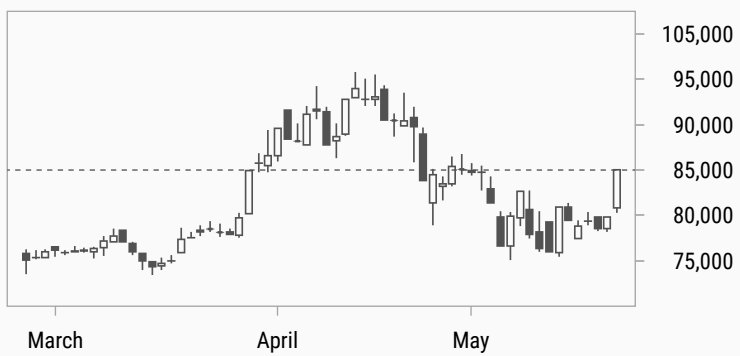
MEAN-REVERSION STRATEGY

Foundation of the Mean-Reversion Strategy

Large corporations in fact cannot change radically in a short time frame such as daily, weekly, or even monthly. However, the market value from stock exchanges is never stable. For many reasons stock prices often fluctuate erratically just within a day. Sometimes they are close to the intrinsic values while other times they are far apart.

Investors pursuing this strategy assume that the stock price will approach the intrinsic value in the long run. Thus when the market price is too low relative to its intrinsic value, investors should open a long position and vice versa.

Figure 04 Stock Price Chart of FPT Corporation



The figure above is an example of FPT corporation's share price for two months, from March 2022 to May 2022. Assuming the business hasn't undergone any major changes over the time period, investors who follow the mean-reversion strategy will look for opportunities through the difference between the stable intrinsic value and 30% moving market price.

Intrinsic Value in the Mean-Reversion Strategy

The mean-reversion strategy is intuitive for many investors. It encourages buying a stock when it's cheaper than its intrinsic value and takes profits on an already overvalued stock.

How does an investor come up with an intrinsic value? It's beyond the scope of this article, though there are theoretical formulas to calculate intrinsic value. In practice, estimating this value requires significant domain knowledge. In this strategy, intrinsic value will shape the investor's perspective on any investment opportunity. The following is a suggested action for the example above of FPT with different estimates on intrinsic value:

Intrinsic Value	Suggested Action
65,000	No investment
85,000	Buy at 75,000; take profit at 100,000
115,000	Buy and hold

Completely different actions under the same market are the effects of estimating different intrinsic values.

In short, to master the mean-reversion strategy, investors need to have skills related to estimating the mean, or the intrinsic value.

Investment Risks of the Mean-Reversion Strategy

Although companies often change slowly, it's not the only factor impacting intrinsic value. Any change in macroeconomics, global markets, government policies, business lines, and competitors can have a major impact on a company's intrinsic value. Misestimating the intrinsic value can lead to buying shares at a too high price while selling shares at a too low price.

The biggest risk in a mean-reversion strategy is that after years of research, the investor may never find the right approach to calculating intrinsic value.

16

EVENT-DRIVEN STRATEGY

Definition

Event-driven strategy is a strategy to find profits from misestimating company prices at major events such as mergers and acquisitions, bankruptcy, share buyback, capital restructuring, extraordinary dividend payments. Investors will study their circumstances and assess how they will impact the company's stock price. They identify potential opportunities that may arise and take advantage of them.

The strategy's effectiveness depends greatly on the investor's ability to capture information, analyze, and evaluate.

More recently, the event-driven strategy has also included pricing after major global macro changes or natural disasters in different regions.

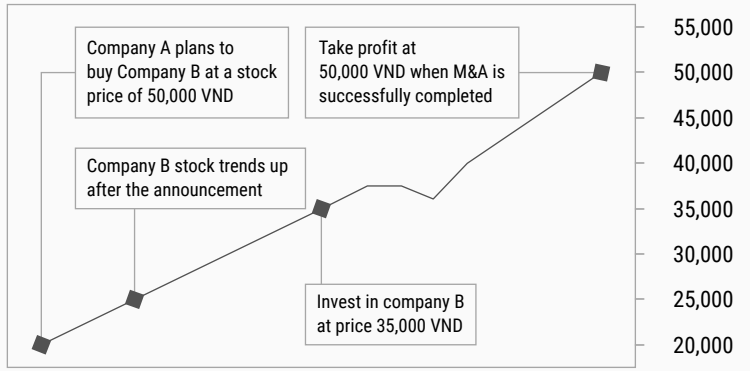
Example of Event-Driven Strategies in Mergers and Acquisitions

Arbitrage trading involving mergers and acquisitions (M&A) activities is one of the popular examples of an event-driven strategy.

Hypothetical scenario: merger and acquisitions with cash payments. Buyers propose buying shares of a target company at a fair price (bid price) and the price is higher than the market price.

Trading strategy: if you believe that the deal will go as announced, you will buy the shares of the target company and sell when the deal closes.

Figure 05 Event-Driven Strategy With a Merge and Acquisition



Example of Event-Driven Strategies in a Global Macro Event

As the Russia and Ukraine war continues to drag on, investors who have information that Russia and Ukraine are major wheat exporters in the world will see a short-term decline in supply. It creates an upward demand for wheat and rice producers. Investors can take advantage of this opportunity to make a profit.

The positive stock movement of Vietnam National Seed Group (HOSE: NSC) in the market downtrend may have partially reflected this event.

Figure 06 NSC Stock Price During Russia and Ukraine War (2022)

Event-Driven Strategic Risk

Incorrect pricing of the event impact is the most common cause of losses for investors using this strategy. In addition, a less common but potentially huge cause of losses is third-party interference. In an event-driven strategy, third parties can be very powerful organizations such as the U.S. or Chinese governments. Actions that can change an investor's entire calculation include police changes, releases of strategic reserves, embargoes, etc.

In addition, there are unexpected internal events. For example, M&A events cannot take place due to a few reasons: shareholders of the target company do not approve, and the security and exchange commission does not approve and does not grant licenses. Another example is that the deal got approved, yet needed time for the completion of legal procedures. When the shares got converted and allowed to trade, the price fell again and investors no longer made profits.

17

MARKET-MAKING STRATEGY

Definition

The market-making strategy simultaneously opens positions at the best buy and sell prices in order to seek profit through the spread. The profitability of market makers comes from the assumption that securities match in both buying and selling directions. Investors may suffer large losses if there's only one-way order execution. It's when they hold too many positions in the opposite direction when the stock price moves in an unfavorable direction.

In Vietnam, market makers are often misunderstood as market manipulators. Manipulators are a group of investors coordinating with each other to continuously buy and sell securities. They create artificial supply and demand, and at the same time spread false rumors. They provide false information to the public, enticing others to continuously place orders to manipulate stock prices.

Market makers essentially benefit from bid-ask spreads. It creates more liquidity in the market. In some markets, this additional liquidity is paid by the security companies or related parties. In most other cases, market makers have enough profits from the spreads.

Market Makers in Vietnam

Market makers in Vietnam can consider two key directions:

- *Participate at the best bid and ask price*. The condition of execution is (sell price - purchase price) > costs. The condition

is rare in the Vietnamese market. However, it happens quite often on days with high volatility.

- *Set an optimal bid and ask price based on algorithms*. In this approach, the market maker will increase the profit margin but the number of matches will be greatly reduced. This is an approach with a higher success rate in Vietnam.

To implement a market-making strategy in Vietnam, investors should note the following two points:

- With the T+1.5 regulation in the underlying security, the market maker needs to have a position available to be able to place a sell order simultaneously with a buy order. With the T+0 trading feature, it's simpler to implement a market-making strategy in the derivatives market.
- Market makers need an automatic, stable, high-speed order system to executive orders, and cancel orders continuously according to each market movement.

18

SCALPING STRATEGY

Definition

Scalping is a strategy that specifically focuses on short time frames for closing and opening positions in order to make small profits. Scalpers often make a lot of trades on a daily basis and expect a very high win rate.

Forms of Scalping Strategy

There are three popular forms of scalping strategy in the market:

- *Scalping as market makers*. Investors order to buy and sell a security at the same time. This action is intended to profit from the difference between the buy and sell prices. This strategy has the best results for stable and high-volume stocks.
- *Scalping as investors*. Scalpers buy in bulk stocks with a strong foundation and wait for the price to move. In Vietnam, with high fees and the T+1.5 rule, scalpers often take a profit when it reaches 2%.
- *Scalping following technical analysis patterns*. In this approach, scalpers wait for a certain pattern to appear to open positions and expect quick profits. This approach often takes profit and stops loss at equal levels. Investors only focus on the profit/loss ratio.

What Investors Need for a Scalping Strategy

In addition to the strategy, investors need a stable platform to place thousands of orders daily. In addition, good data sources are essential to make accurate decisions.

In Vietnam, a stable application programming interface (API) gives investors direct access to a stock brokerage company. It's required for investors who want to execute scalping strategies in the derivatives market.

This strategy brings quick profits so it will attract a large number of investors. However, this strategy is not an easy way to make profits. Investors should note the following important factors:

- Without proper risk management, a large loss may wipe out the profits from hundreds of successful trades.
- Transaction costs are key. With scalping, the profit per trade is usually very thin. It may not be enough to compensate for fees and taxes. Minimizing transaction costs should be taken into account to increase the likelihood of success.
- Liquidity and volatility will greatly affect investment performance. This strategy has high-profit potential when the market sharply fluctuates.
- This strategy takes a lot of time and effort in the case of manual trading.

19

ETF FRONT-RUNNER STRATEGY

Definition

Every public fund must publish an investment prospectus. Index funds must publish details of rebalancing rules. By following these disclosure rules, investors may forecast the next actions of these funds and make a profit by acting ahead.

In the United States, investors following this strategy pocket about $4 billion yearly. The researchers also found most US index funds announce their rebalancing plans in advance. They execute trades at closing prices on index rebalancing days to minimize discrepancies.

The ETF front-runner is a very short-term strategy. To optimize the use of capital, it is necessary to combine it with other strategies. However, this strategy requires constant trading, so fees and taxes can increase multiple times.

In a bearish market, any strategy of taking long positions can incur serious losses. To hedge against market risk, one can consider using a market-neutral strategy. It typically combines buying underlying securities with short-selling derivatives with equal weighting.

Ideas to Apply in Vietnam

Among index funds in Vietnam, DCVFMVN DIAMOND ETF (FUEVFVND) is the best-known fund with the fastest capital growth in recent years. In the DIAMOND ETF prospectus, the

fund's investment objective is to track the DIAMOND index. Therefore, when it's possible to forecast a new list of stocks added to the DIAMOND index, investors can make large profits without much risk. It's because the DIAMOND ETF will likely buy many of these stocks.

Below are a few rules for stocks to be selected in DIAMOND INDEX (2022):

- Have at least 3 months of listing and trading on HOSE as of the closing date of index periodic review;
- Adjusted capitalization free-float value of at least 2,000 billion VND;
- The minimum execution value at 8 billion VND and the trading volume at least 100,000 shares (for shares already in DIAMOND Index from the previous period) or 10 billion VND and 200,000 shares (for shares outside of DIAMOND index from the previous period) to be eligible;
- The number of stocks in the index is at least 10. FOL of VNDiamond stocks in the previous period is at least 80%, where FOL is defined as:
  FOL = Foreign investor ownership/Foreign investor holding limit
- Stocks in the previous period with P/E at most 3 times the average P/E. For stocks outside of DIAMOND index, P/E is at most 2 times.
- The proportion is at most 40% for a single industry.

By statistically using the complete set of rules in the prospectus, investors can accurately predict the list of stocks in the DIAMOND Index. Large securities companies like SSI JSC, or VNDirect JSC often make their own predictions. Following and forecasting their reports in advance will optimize profits for a trading system.

20

ARBITRAGE STRATEGY

Definition

Arbitrage is a strategy that takes advantage of temporary price differences of the same asset class in two different markets. It trades and takes profits without incurring much risk. Investors implement arbitrage strategies by buying assets in a lower-priced market and simultaneously selling them in a higher-priced market.

Arbitrage strategies are closely related to efficient market theory. This theory states that the market is perfectly efficient when all past and present information regarding the asset is quickly and reasonably reflected in the market price. In other words, efficient markets will allow no opportunity to take advantage of arbitrage trading.

However, for many reasons, subjective or objective, there are always periods of inefficiencies in the real market. That's why the arbitrage strategy is widely used, and probably one of the oldest trading strategies in existence.

Two Main Characteristics of Arbitrage Strategies

Only trade when there's an imbalance in asset price. This is the most important condition of arbitrage trading, which can take several forms:

- The same asset but being traded at different prices on two different markets; or
- Two assets have similar future cash flow expectations but are trading at different prices.

Only trade simultaneously. The buy and sell transactions must take place at the same time. The investor may be exposed to significant risk if there is too much difference in the timing.

Arbitrage Trading in Practice

Arbitrage strategies almost don't exist in the Vietnamese stock market. Arbitrage opportunities are more evident in the foreign exchange and cryptocurrency markets and in stocks traded over two markets, like the US and European markets.

In modern finance, arbitrage opportunities often do not exist for individual traders but only for international banks. They can continuously use and convert different foreign currencies.

An example in the modern economy of arbitrage is Bitcoin in South Korea versus the rest of the world. In 2017, the bitcoin price in South Korea was 50% higher compared to the rest of the world. Individuals/organizations who can buy and sell continuously legally may make extremely large profits without risks. However, most were unable to take advantage of this one-off opportunity although they were aware of it.

In Vietnam, the cooperation between the State Securities Commission of Vietnam (SSC) and the NYSE to develop a mechanism for investors to participate in the two stock markets was signed in May 2022. This can open up arbitrage opportunities in both markets. It is a potential project for small investors although major banks around the world dominate the field.

Barriers to Entry in Arbitrage Strategy

The development of trading systems, along with algorithmic trading leads to the extremely short occurrence of arbitrage trading opportunities. Sometimes they can last for just a few seconds. In this context, the traditional method of placing orders

will find it difficult to seize the opportunity and trade on time. However, in terms of the overall market, the more the transaction speed improves, the more efficient the market will be.

Despite arbitrage trading opportunities, the price gap is also a key factor. An investor only makes a profit if the price difference can cover the trading fees and costs.

21

GRID STRATEGY

The grid strategy can make a profit in an oscillating market. However, it may have a big risk in trending markets.

Definition

Grid strategy sets up a continuous price grid of buying and selling at predetermined prices to gain profit in any market. This strategy works best in an oscillating market with large swings.

The following is an example of a neutral grid setup on FPT corporation's stock: the starting reference price of the grid is 87,200 VND. The grid step is 800 VND and the take-profit price step is also 800 VND.

Price	Position	Take-Profit Price	Profit
89,600	Short	88,800	800
88,800	Short	88,000	800
88,000	Short	87,200	800
87,200	N/A	N/A	N/A
86,400	Long	87,200	800
85,600	Long	86,400	800
84,800	Long	85,600	800

Grid Type

- *Neutral grid*. This grid works best in oscillating markets. The grid starts with zero positions and accumulates long positions if the market goes down, and short positions if the market goes up.
- *Uptrend grid (bull market grid)*. This grid works best in bull markets. The grid starts with several long positions instead of no positions.
- *Downtrend grid (bear market grid)*. This grid works best in bear markets. The grid starts with several short positions instead of no positions.

Important Parameters

- *Trading price range*. The highest price and lowest price of the grid.
- *Grid*. The distance between each pending order and the profit/mesh. This distance can be between each pending order and can be a fixed number (arithmetic grid) or a fixed percentage (geometric grid).
- *Stop-loss*. Does the system allow stop losses and when will it be activated?

Risks

Grid strategies work best in oscillating markets. Any trending market can cause this strategy to accumulate a large number of positions against the trend. It may cause heavy losses to the entire account.

In a bull market, a neutral grid will take short positions and accumulate more and more at a new high. If a stock constantly reaches new highs, the neutral grid can consume way more positions, leading to serious losses.

Application

The grid strategy can make significant profits in an oscillating market. It's vastly different from the momentum approach. Any trading instrument can use the grid trading system. Also, because of its simplicity, the grid strategy can be easily automated. Today, the grid strategy is most popular in the currency and cryptocurrency markets because of its 24/7 trading nature. It's very advantageous for the grid strategy. In Vietnam, a grid strategy can also be developed. However, risk control is key due to the extremely high volatility of small markets.

22

SMART-BETA STRATEGY

Definition

Smart-beta strategy, or factor investing, is a strategy for building a portfolio according to a system. It's based on rules, and the use of fundamental business factors, e.g. liquidity, value, and quality, as criteria for making investment decisions.

Smart beta is the scaling strategy of the passive investment strategy. With passive investing, investors only need to buy the entire component shares of an index as a proportion of market capitalization. They hold for the long term to achieve returns comparable to the reference index. With smart beta, the portfolio weights are determined based on fundamental factors rather than market capitalization. The goal is to increase the proportion of stocks with high return potential, and in contrast, reduce the proportion of less potential stocks. It optimizes the portfolio with better returns compared to passive investment strategies. Theoretically, smart beta combines components of both passive and active investment.

Example of Smart Beta

An investor believes that over the long term, stocks with low P/E tend to generate higher returns than the overall market average. Applying this investment perspective by taking VN-Index as a benchmark, this investor builds a portfolio according to the Beta strategy as follows: buy all VN-Index shares and hold for the long term. The weights are determined according to the rules: the lower the P/E, the higher the weight.

As an illustration, assume VN-Index has only 5 component stocks A, B, C, D, and E. Investors use P/E as a parameter to calculate the portfolio weight as shown in the table below:

Share	P/E	Portfolio Weights
A	5	15/50 = 30.00%
B	7	13/50 = 26.00%
C	10	10/50 = 20.00%
D	13	7/50 = 14.00%
E	15	5/50 = 10.00%
Summary	**50**	**100.00%**

This strategy is quite similar to passive investing, but the P/E ratio is used to determine portfolio weights, instead of market capitalization.

In the same instance, investors can determine portfolio weights using other methods, as long as the first principle is maintained: "Increase the weights of stocks with low P/E and reduce the weights of stocks with high P/E." Below are some prime examples.

- Only invest in x stocks in VN-Index with the lowest P/E, also known as shortlisting. For example, suppose VN-Index has 500 stocks, invest in only the bottom 50 stocks with the lowest P/E. The proportion is 2% for each shortlisted stock and none for others;
- Determine an acceptable threshold of the P/E and only buy stocks satisfying this constraint.

Some Commonly Used Factors in Smart-Beta Strategy

In investing, a "factor" is understood as a variable or characteristic highly correlated with stock returns. The factor can also be defined broadly as any variable that investors believe is valuable in ranking stocks and predicting future returns and risks. Below are 5 commonly used and widely accepted factors.

- *Value*. In the long run, investing in stocks undervalued by the market has a higher rate of return than overvalued stocks. The value factor can be measured from key metrics in corporate financial statements, such as P/E ratio, dividends, income, cash flow, EBIT, and EBITDA.
- *Market Capitalization*. Small-cap stocks tend to rise more sharply than large-cap stocks.
- *Volatility*. Increasing the weights of low-risk stocks can help improve the portfolio's return on risk.
- *Quality*. High-quality companies will increase the share price over time to reflect their intrinsic value. Some basic criteria of high quality are strong, stable profitability, and low leverage ratio.
- *Momentum*. Recently rising stocks tend to continue gaining similar momentum in the short term.

In reality, there are hundreds of factors that have been identified and used in building portfolios under smart beta. Additionally, investors can combine them to create more variations of this strategy.

Strengths and Weaknesses

Combining active and passive investing, the smart-beta strategy has several advantages of passive investing such as transparency, transaction costs, and low monitoring costs. At the same time, smart beta also shares advantages of active investing like a higher

potential return than average, and diversified investment depending on the preference and domain knowledge of each investor.

However, identifying key factors and rules in smart-beta strategy can be complex. It requires skills in testing historical data and experience, plus in-depth knowledge of financial analysis. Besides, these factors often come from corporate financial statement data. In Vietnam, it can be a huge hurdle to gather detailed, complete, and accurate financial statements of all listed companies. It remains the main challenge for investors following this strategy.

Application in Algorithmic Trading in Vietnam

With an algorithmic trading system, computing the financial factors of all stocks in the market has become simpler than ever. Besides common metrics like P/E, P/B, and ROE, investors can freely customize further metrics as desired. An example is the short-term unrealized revenue divided by the total revenue of the last 4 quarters. It reflects the percentage of potential revenue recorded in the next 4 quarters. Under the same conditions, the higher this factor is, the more stable and solid the company is.

An algorithm following a smart-beta strategy can include many of the customized metrics like those above. They can be adjusted to different weights as the portfolio desires.

Note that it's key to restrict frequent portfolio rebalancing and aggressive trading. It's to ensure the passive investing components of the portfolio.

23

SNIFFING STRATEGY

Definition

"Sniffing" is a special group of algorithms to detect other trading algorithms. The sniffing strategy usually aims at continuously high-volume trading algorithms like TWAP, VWAP, and POV. It can also detect market-making algorithms or price momentum algorithms albeit less commonly used.

The input data of sniffing algorithms includes real-time tick data with the order book of all traded securities.

The key principle is to rely on matching orders and order books to find patterns that are more repetitive than arbitrary. Once the predictive model has been identified, if the above pattern continues, the sniffing strategy is considered to have identified another algorithm. Depending on the obtained data, it can mirror to open similar positions to take advantage of the information. Trading algorithms appear in different forms below:

- *TWAP*. The main characteristic of the TWAP algorithm is that the orders are separated over a fixed period of time. Sorting trading orders and finding transactions with similar volumes separated by a constant time period is a strong indicator of TWAP.
- *Order splitting*. The characteristic of order splitting is that many orders match in a brief time frame of under one second. A simple model reporting the number of executed orders over a 5-second period can give a strong signal of order splitting.
- *Market making*. The market-making strategy usually works very strongly at the bid price 1 and the ask price 1. All

securities that continuously place orders at this bid and ask price usually involve a market-making algorithm.

- *Price momentum*. Price momentum algorithms are often used in the derivatives markets. The common characteristic is to buy when the market is trending up and sell when the market is trending down. It usually creates a large price slippage. Monitoring repeated large slippage points over several days can give a reliable signal of high-volume price momentum algorithms in the market.

Key Applications

When detecting possible transactions of large organizations, the basic principle is to take positions in the same direction at a reasonable volume. It's key to note that price changes too quickly can change the strategies of investors who are accumulating positions. So opening positions need to be strategic and rational according to the general market. Investors need to consider taking profits when the price moves too fast, or there's no longer a support signal from the detected algorithm.

When detecting stocks with the participation of market-making algorithms, applying the right price momentum strategy at the right time can bring great profits.

When detecting a price momentum algorithm, using a scalping strategy will be optimal at points where algorithms with large volumes are involved.

Protect Your Systems From Sniffing Algorithms

Stealth algorithms are a group of algorithms to help execution algorithms avoid detection from sniffing algorithms. The basic principle of sniffing algorithms is to detect patterns. Adding more randomness in the execution will likely work well in defending sniffing algorithms.

24

HIGH-TECH ALGORITHMIC TRADING

In global financial markets, any approach that brings high potential profits is of significant consideration. It includes the most modern technologies that may give investors a strong edge.

Satellite Images

Imagine you know exactly how the Russia-Ukraine war is unfolding in real time. You stay up-to-date on all Russian tankers and the current state of Ukrainian wheat production. With this information, how possible is it to make big profits with low risks? Does this sound like fraud to ordinary investors?

There are many satellite image providers in the market such as RS Metrics, Descartes Labs, Orbital Insight, and Planet. This proves to create a huge competitive advantage for clients of these companies during global and macro events. Having this information just one second before the news announcement can potentially bring large profits.

In practical applications, automobile counting is the most basic approach used to assess the potential of retailers. Other approaches include counting solar panels, timber inventories at sawmills, and counting metal mining vehicles worldwide.

Although promising, it's currently not feasible in Vietnam because the market size is very small. Satellite imagery is unlikely to add much value. Besides, Vietnamese investors cannot, due to laws, invest in foreign markets where this approach can be of good use.

Sentiment Analysis

At 1:07 pm on Tuesday, April 23, 2013, the Associated Press Twitter feed stated Barack Obama had been injured in an explosion at the White House. The stock market fell 0.9% immediately in just a few seconds. After confirming the tweet is fake, the market has fully recovered. It's still a vivid example, however, of how sentiment strategy works in real trading.

In principle, a sentiment strategy is an approach that uses words from global news, such as social accounts of supposed credible sources. They are to capture the latest news or the crowd mentality on a topic. From the data, the algorithm automatically opens the corresponding trading positions. Investors interested in this approach may consider Sentifi solutions.

At the time of publishing, there's no evidence that sentiment-driven algorithmic trading has a competitive advantage in Vietnam. Still, this approach is a compelling and intriguing idea to follow.

Machine Learning and Artificial Intelligence

This approach is probably the most well-known for its applicability in various fields. However, in the financial sector, not many machine learning or artificial intelligence algorithms are known to be sustainably profitable. It is due to the random nature of the financial sector, which can be very different from other scientific and rule-based domains. In random datasets, many models that perform well with the past data will not guarantee future performance. This fact can make the trading performance of machine learning and artificial intelligence models ineffective. However, in the future, artificial intelligence will still dominate the trend in financial market research as we collect more and more data.

25

BEHAVIORAL FINANCE IN ALGORITHMIC HYPOTHESIS FORMATION

Definition

Behavioral finance studies how people make decisions and examines the psychological factors that influence and skew their decision-making process.

The traditional economic and financial theory generally assumes that market participants always act rationally by considering all available information in the decision-making process. In reality, financial decision-making is a complex situation. When faced with so much information to constantly process and update, people often don't have the time or ability to come up with a completely optimal decision. Instead, they often take an easier and more subjective approach. They often use only parts of the available information and determine a course of action that best suits their judgment and priorities. They are content with making a "sufficiently good" choice rather than making an "optimal" choice. In this way, they may inadvertently bias the investment decision-making process.

Behavioral finance does not assume people are always rational. It believes that people are limited in controlling themselves and are influenced by their own biases. These biases are in two main groups: cognitive errors and emotional biases.

Cognitive Errors

These types of errors are basic statistical errors, information processing errors, or memory capacities. They cause decisions to deviate from rationality. In general, cognitive errors stem from faulty reasoning. It can often be, however, mitigated or eliminated through better information, education, and advice. Cognitive errors can be grouped into 2 types:

- **Belief perseverance biases:** come from the result of mental discomfort. It occurs when new information conflicts with pre-existing beliefs or perceptions. To resolve this dissonance, people are likely to ignore conflicting information. They only consider information that confirms their existing beliefs and thoughts.
- **Information processing biases:** refer to information being processed and used in illogical and irrational manners.

Below are some common belief perseverance biases and their corresponding behaviors in financial investment.

Bias	Consequences
Conservatism Bias Tend to hold on to past information.	■ Maintain existing views and predictions. Be slow to change even when market information has changed. ■ Maintain existing views when facing difficulties in processing new and complicated information.
Confirmation Bias Tend to only notice what confirms previous beliefs, underestimates or ignores anything that contradicts it.	■ Only consider positive information about the existing investment and ignore any negative information about it. ■ When building a portfolio, investors may have been convinced of the value of a certain company's stock. They may have ignored the negative news, gathering only information that confirms the company is a good investment. They tend to give more weight and hold fewer diversified stocks. ■ Hold a high proportion of the shares of the working company. At the same time, only talk about the company's good news and ignore unfavorable information.

Bias	Consequences
Illusion of Control Bias Tend to believe one can control or influence outcomes when in reality they cannot.	■ Investors believe they can have control over investment outcomes, which may lead to excessive trading. ■ Investors prefer to invest in companies they may feel they have control over. One example is the one they work for, especially in a high position.
Hindsight Bias When the event has already happened, one looks back and assumes that they already predicted it correctly.	■ Investors overestimate their ability to predict future investment results. For example, an investment has profits for unforeseen reasons. However, investors may rewrite their own memories to explain how they were able to predict this profit. It may be different from what actually happened. This happens because when people look back, they don't have a perfect recollection and tend to fill in the gaps with what they want to believe. ■ Tendency to rewrite the past can lead investors to take excessive risks. This may lead to future fatal mistakes.

Below are some common processing errors and their corresponding behaviors in financial investment.

Bias	Consequences
Anchoring Bias Tend to rely on parts of original information to make further estimates, judgments, and decisions.	■ Investors may get too attached to the original estimate and make inadequate adjustments even when new information is updated.
Mental Accounting Bias Divide the money into different accounts mentally and affect the investment decisions.	■ Investors divide the portfolio into several parts with different risk levels. Each part is designed for a specific purpose. This approach neglects opportunities to mitigate risk by combining assets with low correlations. ■ Investors consider principal and profit as separate parts. They are willing to use the profit part to trade with higher risk.
Framing Bias Answer a question differently depending on how it is framed or structured.	■ Focus on short-term price movements. Ignore long-term considerations in the decision-making process.

Bias	Consequences
Availability Bias Estimate the probability of an outcome or a phenomenon based on how easily information is obtained.	■ Choose an investment based on advertising and recommendations of many people. Ignore a thorough analysis of options. ■ Invest in a company only because it suits personal preferences or because of familiarity. Ignore any risk and return evaluations. ■ Select stocks based on limited experience. Investors who are working in the real estate industry, for example, tend to allocate most of their portfolio weights to real estate stocks. Though having lots of relevant information and experience in their field, they fail to diversify and achieve an appropriate asset allocation.

Emotional Bias

Emotional bias arises spontaneously from feelings and attitudes. It can cause decisions to deviate from rationality. It is typically harder to correct compared to cognitive errors because they originate from impulses and intuition rather than logical calculations. A person usually can only recognize and learn to adapt to it rather than correct itself.

Below are some common emotional biases and their corresponding behaviors in financial investment.

Bias	Consequences
Loss-Aversion Bias Consider losses more serious than equivalent gains.	■ Find it difficult to accept stop-loss. Hold losing positions unreasonably longer in the hope that the share price will return to break even. ■ Sell winning investments too early for fear that profits will decrease.
Overconfidence Bias Express an unfounded belief in one's own abilities	■ Underestimate risks and overestimate expected returns. ■ Hold under-diversified portfolios that can lead to significant risks.
Self-Control Bias Unable to pursue long-term goals because of the lack of self-discipline.	■ Investors have difficulties saving sufficiently for the future by sacrificing current consumption. When they realize this, they may take excessive risks to generate higher returns.
Status Quo Bias Choose to do nothing instead of making a change.	■ Investors may unknowingly maintain portfolios with inappropriate risk characteristics. ■ Fail to explore other investment opportunities.

Bias	Consequences
Endowment Bias Tend to value an asset more when they hold rights to it rather than when they do not.	■ Fail to sell off certain assets. Thus maintain an asset allocation inconsistent with the investor's risk tolerance and financial goals. This is especially true for inherited investments. When investors inherit a number of stocks, for example, they fail to sell off even when these stocks face a bad outlook due to emotional attachment.
Regret Aversion Bias Avoid making decisions for fear that decisions will turn out poorly.	■ Be too conservative in investment choices due to poor past results. ■ Engage in hedging behaviors. Investors feel safer in popular investments.

Applications of Behavioral Finance in Algorithmic Hypothesis Formation

The focus of behavioral finance is the study of behavioral biases that influence individual investor decisions. Behavioral finance doesn't show how to properly predict the future. Nor does it provide a financial investment model that can confidently beat a market. However, behavioral finance still helps investors have a deeper insight into how the market actually works. It helps build more reasonable algorithmic hypotheses.

Price momentum is a common phenomenon in the stock market. It's an unusual movement of the stock price. It can be

observed that when a stock rises or falls sharply in the short term, the trend tends to continue.

In terms of behavioral finance, price momentum can be explained by availability bias – the tendency to forecast an outcome or the importance of a phenomenon based on how easily the information is recollected.

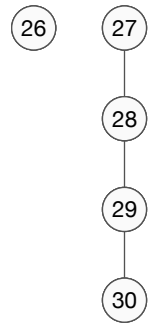
When a stock price keeps rising, investors easily remember this event and unconsciously assume a future rise is more likely than a fall. Besides, hindsight bias causes illusions to investors who believe they could have forecasted a future rise and would regret it when missing the buying opportunity. Investors are thus urged to overcome this regret by placing buy orders even when the stock's market price is already much higher than its intrinsic value.

Believing the biases above can have a strong market impact, algorithmic traders can take advantage of the price momentum strategy to seek short-term profits.

Behavior finance also explains the phenomenon of absurdly sharp rises in stock prices during a brief period of time. In addition to the price momentum insight above, buying and selling shortly after will still be profitable even if investors have sold too early. A few short-term profits bring a sense of pride and confirm the investors' belief that they made the right investment.

Hindsight bias can reinforce an investor's belief that their previous trading strategy was correct and will continue to be profitable in the future. Overconfidence bias gives an illusion of their forecast ability. Conservatism bias makes investors ignore contradicting information or opinions. As a result, investors are still willing to open more long positions at high prices, even using margin to increase purchase power. It further causes increased demand and the stock price continues to rise rapidly.

Yet in the context of absurdly rising stock prices, investors can apply the mean-reversion strategy. They can open short positions when the security price already rises too quickly and expect it will revert back to the reasonable mean.



CHAPTER III

DATA

26

STANDARD DATA IN ALGORITHMIC TRADING

Market Data

Market data consists of the following essential data fields:

- Stock ticker symbol;
- Order execution time;
- Order execution price;
- Order execution volume;
- Bid price 1, 2, 3 (10 price steps for HNX and UPCOM);
- Bid volume 1, 2, 3 (10 price steps for HNX and UPCOM);
- Ask price 1, 2, 3 (10 price steps for HNX and UPCOM);
- Ask volume 1, 2, 3 (10 price steps for HNX and UPCOM).

These data fields allow investors to use most technical analysis strategies or OHLC charts (Open-High-Low-Close). This is the simplest and most common data group sold under data packages in Vietnam.

Execution data is also categorized based on the type of the transaction such as insider transaction data, foreign investment execution data, and deal execution data.

Insider transaction data is publicly available. However, in Vietnam, insiders may trade in different ways to avoid affecting stock prices and frequent reporting. This data is therefore often unreliable, and it should only be used as a reference.

Foreign investment data represents a strategic group of shareholders. This group plays an important role in making long-term investment decisions. However, it is not as useful for short-term investments.

Deal transaction data has a high volume and reflects the expected prices of the parties involved. In special cases, when foreign investors cannot buy shares directly on the stock exchange, the deal transaction often represents the valuation of foreign investors. However, the public information of the deal transaction may still differ from reality. This group of data is not completely reliable in the Vietnamese market.

Financial Statement Data

This data group contains:

- Income statement;
- Balance sheet;
- Cash flow statement (direct or indirect).

This data group is widely used in fundamental analysis. However, Vietnamese accounting standards still have many differences from the international standards. Different industries such as manufacturing, banking, securities, and insurance have specific financial report structures. To analyze the whole market, it's essential to have reference benchmarks.

Dividend Data and ESOP

The company's profit-sharing policy and the ESOP program over many years have greatly influenced investor sentiment in the long term. Therefore, data including cash dividends, stock dividends, and bonus shares can be used in algorithmic trading.

ESOP is a special policy to attract corporate talent. However, it's often abused for personal purposes. When adopting the IFRS, international financial reporting standards, ESOP abuse in Vietnam should be more restricted.

Macroeconomics Data

Inflation data, interest rates, economic growth, import and export, exchange rates, export orders, money supply, and public investment reports are publicly available macro data. They are often used to predict the economy in the medium term.

For instance, algorithmic traders can rely on inflation and interest rate data of 2022 to gain insights into the macroeconomic impact on the market and domain-specific businesses.

Commodity Data

Each particular industry and enterprise will be closely related to the price of imported and exported goods. These prices often move together with the global commodity prices, which are important data to collect. Investors in Vietnam are most interested in common commodities like oil, gold, soybeans, coffee, and meat. Other commodity prices are more complex but still collectible such as international freight shipping prices.

During the Covid pandemic, the supply chains were disrupted. Commodity prices no longer followed normal economic rules. They become important information for seeking trading advantages.

Index Data

A single stock or a particular industry usually has a certain correlation with the local and global market indices. Therefore market indices need to be stored to have an accurate view of the Vietnamese market or each individual stock in the global context. Besides the data above, there are other types of specialized data. Algorithmic traders can gather them to support algorithm development and seek trading alpha.

27

DATA CLEANING TUTORIAL

Clean, accurate, and complete data is the foundation of an algorithmic trading system. Inaccurate data will affect the performance of the algorithm during the testing phase and real trading.

Common Data Errors

- *Data collection programming errors*. Common in internal data collection systems and third-party data services.
- *Partial data loss errors*. Common when the data collection system and storage experience unexpected problems. For example, data sources changing their data structure will lead to data loss errors.
- *Raw data errors*. Occur when the data supply for the whole market experiences issues that cannot be fixed in a short term. An example is an erroneous buy order for tens of millions of contracts on the Vietnamese derivatives market.
- *Manual input errors*. Occur on financial statement data and other information that has not been automated at the data entry step.
- *Empty data field*. Appears when scanning is incorrect on financial statements of stock tickers on UPCOM for example.
- *Time difference*. Occurs when data sources update with a large delay compared to the time stamp issued. This error is most common in financial statement data of small companies in the UPCOM market.

- *Past data adjustment*. This is a particularly difficult error to fix when past data is adjusted for various reasons. This error is common in stock dividend data.

In Vietnam, data errors outside the price and volume range also often occur. Algorithmic traders should carefully consider data sources before developing algorithms. For inaccurate and skewed data that cannot be automatically collected, traders should consider skipping this data source until the data cost falls to an acceptable level.

How to Clean Data

Standardize Data

- *Define data standards*. Different systems will always have different data standards. However, data needs standardization to ensure compatibility, ensuring data retrieval tasks, and system scalability in the long run. For example in a Python system, the time standard is the float data type in Unix timestamp.
- *Standardize data*. Input data typically has different formats compared to those in algorithmic trading systems. It's important to standardize data before system integration.
- *Define the trusted data source*. This data source is the most trusted source and can provide first-tier data to the entire market. Based on the price and volume data of Vietnam's derivatives market, HNX is the trusted base data source. The base data source typically has the highest accuracy and stability. Defining and using the trusted base source will enhance the stability and reduce overhead costs of data cleaning.

Data Validation

Compare two data sources to validate and clean data. Basic validation techniques include:

- *Identify missing data*. Add the missing data to the system database or remove it entirely.
- *Identify data duplicates*. Remove duplicates. Duplicate data is usually related to a listed company event and is often published at different times.
- *Identify data anomalies*. Remove any anomalies. Anomaly data can be detected using a simple statistical and probability method. An example of data anomalies in the Vietnamese stock market is the unchanged derivative price, as the trading volume exceeds the processing capacity of the trading system. It results in a constant price over a long period of time. Another example is the occurrence of unusually large orders at millions of contracts, while the average daily trading volume is around 200,000 contracts.
- *Identify invalid data*. Trading prices that exceed ceiling and floor prices or other straightforward constraints are clear signals to review and clean up data. Typically an invalid data point will help identify a range of data that needs cleaning and adjusting.

In addition, some data errors are easy to recognize with a close look. An example is the closing volume in the Vietnamese stock market may not be updated correctly by third-party service providers.

28

DATA MANAGEMENT IN ALGORITHMIC TRADING

This section covers different data aspects in algorithmic trading. We will discuss the two groups of data and principles of data collection, as well as common data management issues.

Two Data Groups in Algorithmic Trading

There are two main groups of data in algorithmic trading: input trading data (market data, financial data, commodity data) and output trading data (algorithmic trading data). Sometimes, the output trading data may serve as input for other algorithms.

- **Input trading data** includes but is not limited to standard data in algorithmic trading as described in article 26. An example is real-time market data of a financial instrument. Real-time data can run at different speeds depending on the market size.
- **Output trading data** is the data generated when trading systems operate. They can be both buy and sell signals generated based on input trading data or algorithm trading orders. While the input trading data is to operate algorithms, the data generated during operations is to monitor the algorithms. Besides, this data group is also used to research and develop new algorithms. During operations, it is key to keep only important information so as to not overload the system operator as well as to optimize data storage.

Two Important Criteria in Selecting Data Sources

Data latency and completeness are two main points when selecting data sources. Meanwhile, defining key concepts and mastering technology is of utmost importance in collecting and managing data.

Data Latency. Latency is the time difference between when data is generated versus when an algorithm receives it. Latency is a key aspect of real-time trading, regardless of market size. Investors need to be concerned about data size as the market grows. However, latency always has a higher priority. The reason is in both large and small markets, if the data latency is very large, the trading algorithm will not be able to take advantage of market opportunities. However, in some trading data, algorithmic traders may tolerate higher latency than usual.

Data latency can be minimized by accessing the best real-time trading data source. Since latency is the top priority when choosing a data source, it will lead to the critical choice of the physical location of a data collection system or a data server. Due to regional and service limitations, investors should anticipate technical problems when setting up and maintaining services. It also optimizes costs when implementing trading algorithms in the future.

Data Completeness (Data Coverage). Standard data completeness or coverage is the ratio of data obtained from a source to actual data. For example, a market has a total trading time of 360 minutes. The data source, due to technical limitations, can only record 336 minutes of trading data. In this case, the completeness of the data source is 336 divided by 360, which is approximately 93.33%. In most cases, it's not straightforward to measure the coverage of transaction data in real time. Therefore,

having multiple data sources and cross-checking with each other is highly importance.

Data Storage and Management in Algorithmic Trading

Key Concept. The first thing to keep in mind is not the storage system but what data should be stored. Algorithmic traders need to identify what data to store. Some basic concepts right from the beginning are prices, orders, signals, and order execution. Selecting the right concepts helps in planning and designing a more efficient data system. A reliable database contributes to fast, smooth, flexible, and scalable system development. This also makes the data storage system easy to upgrade in the future.

Technology Selection. After defining concepts and data fields, the choice of the technology stack is not as important. The right tools and skills in technology are more important than the technology choice itself.

Data Storage Management. There are two main types of data storage in algorithmic trading: temporary storage for data trading, and database for historical trading data.

- *Temporary storage*. This is a repository of real-time trading data, including market data, index, and commodity data. Since algorithms must reach data as quickly as possible, storing data in an in-memory database, or caching system like Redis is a reasonable choice. Note that the data should have an expiry time. It's because this data is often used for intraday trading. Temporary storage data from the previous day, or any previous period, will often be unusable. Using improper data can lead to serious consequences, especially for intraday trading algorithms. Therefore, checking data correctness in terms of time, and determining expiry times

are mandatory. If data from previous days or periods needs to be accessed, a database is more suitable than a temporary storage system.

- *Database.* When a trading day ends, the intraday trading data should be written down in the database for long-term use. This is to research and develop new algorithms, backtest existing algorithms, and improve operating algorithms. Many types of databases are popular like Postgres, MySQL, etc. These databases are used by data management and mining tools like ElasticSearch, Logstash, and Kibana (ELK) to display, illustrate, track, report, summarize, analyze, and evaluate historical data.

29

STOCK TRADING API IN VIETNAM

API Definition

An **application programming interface** (API) is a set of definitions and protocols that allow two software to communicate with each other.

To develop an algorithmic trading system, investors need to optimize the process of obtaining data, retrieving positions, and placing/editing/canceling orders through APIs provided by securities companies or exchanges. In Vietnam, the process is implemented with a frequency of approximately one cycle every 02 seconds. In developed markets, the frequency can be thousands of times faster. Investors need programming experience to call API properly, including authentication, key retrieval, connection configuration to the server, sending queries, parsing response packets, performance testing, and security testing.

API Categories

- The REST API only returns data when requested.
- The FIX API returns data immediately upon data update.
- The RPC API returns data immediately upon data update.

With these three API categories, there's the speed superiority of FIX and RPC APIs over the remaining.

Stock Trading API in Vietnam

A pioneer in providing open APIs for the investor community is the Bank of Investment and Development of Vietnam Securities Joint Stock Company (BSC). DNSE Securities JSC focuses intensively on supporting algorithmic trading through API as well as AmiBroker analysis software. More recently, SSI Securities Joint Stock Company has just launched an API service. With its leading position in the industry for many years, it has marked the official period of widely accepted algorithmic trading in Vietnam. Their detailed API information is available publicly online. Other securities companies have also fully developed their APIs but have not announced them officially yet.

Which API to Choose

- *SSI and BSC APIs are suitable for beginner traders who need to familiarize themselves with algorithmic trading and require stability*. In the introductory stage of algorithmic trading, a stable and simple-to-use API with a professional support team is key. BSC has many years of experience in implementing API services. SSI with Q-Trader trading software also meets the needs above.
- *DNSE is suitable for high-volume experienced traders*. Apart from basic functionality, high-volume traders are most concerned with costs. With the direction focusing on the algorithmic trading market, DNSE is a stock company that meets this criterion when trading large volumes.

Over time, new APIs will roll out with many improvements. Currently, API development is no longer a technological challenge for securities companies. Many big names have joined the market. Traders can expect that big securities companies

such as HSC, VND, and VPS will soon announce their own APIs, thereby adding options to the market.

Investors in Vietnam have witnessed a profound quantitative and qualitative shift in the transition from trading over the counter to online trading. A bigger revolution for Vietnamese investors will be from online trading to automated, algorithmic trading.

30

SEARCH PROCESS FOR FASTEST DATA SOURCE

Higher Speed, Higher Efficiency

Algorithmic traders all want access to the fastest data source to improve performance while reducing transaction randomness.

For some algorithms like market-making, scalping, and arbitrage trading, the fastest data source means a lot of real profits. A slower data source only leads to paper profits.

In other algorithms, the fastest data source plays a role of reducing noise. Reducing noise will ensure algorithm performance isn't the result of randomness, even though noise is not inherently an adversary for trading.

Finding the fastest data source is therefore in high demand for all algorithmic traders, especially for the Vietnamese stock market.

Available Sources in Vietnam

There are four main data provider groups, each consisting of multiple data sources:

- *Live board of security company*: this is data from live boards of brokerage companies like SSI and VND. There are at least 50 data sources in this group. Some require registration while others offer it for free.

- *News website*: data from financial news websites like Cafef, Vietstock, and Stockbiz. These websites provide free data, but do not aim to serve traders, so the latency is very high.
- *Data service provider*: data from data providers such as Fialda and Fireant. AmiBroker is popular among Vietnamese algorithmic traders. It is one of the most popular data sources.
- *API*: the application programming interface provides free data. However, to access data from the API, the algorithmic trader needs programming skills.

The Speed of Data Group in Vietnam

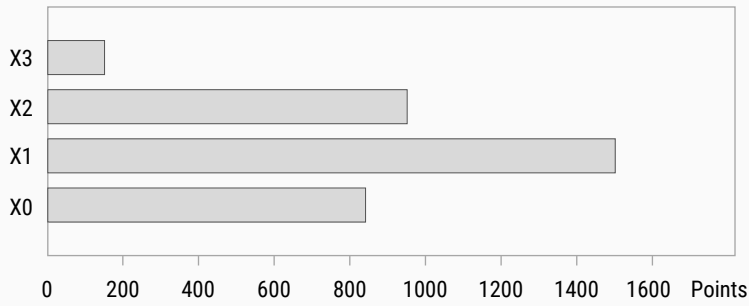
The ranking of each data group is as follows:

- API is the fastest group. It's built exclusively for algorithmic trading from brokerage firms. It's always the fastest data pool in the market.
- Data service provider is ranked as good. Data needs standardization and moves through multiple intermediary stations before reaching the end user. This can cause some latency.
- Live board of security companies is ranked as average only. It only focuses on user experience rather than speed. It definitely does not have the top speed.
- News websites are the slowest. It aims to provide financial news, and speed isn't among their priorities.

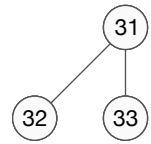
Verify the Fastest Data Source in Vietnam

We first collect all the FIX APIs and RPC APIs available in the Vietnamese stock market. We use a simple algorithm to count the number of times each API is fastest. The API with the highest points is the fastest data source in the market. The figure below tests API speed on Feb 14, 2023.

Figure 07 API Speedtest



At ALGOTRADE, the daily score is used to determine the current fastest data source. It will automatically be used as the main data for the next trading day.



CHAPTER IV

BACKTESTING

31

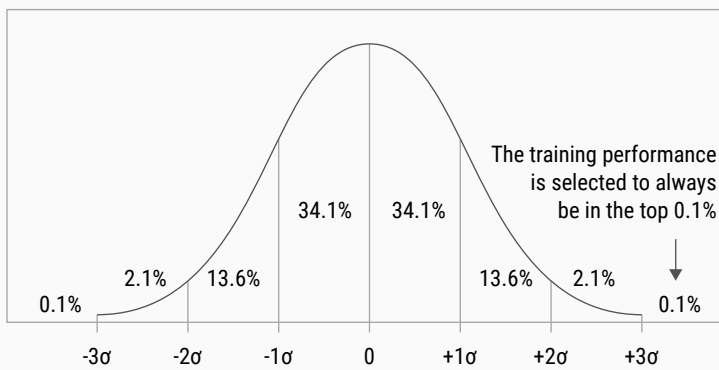
PHILOSOPHICAL FOUNDATIONS OF BACKTESTING

Investors often try hard to experiment with a lot of third-party software. They may find many potential trading algorithms but end up losing hard-earned money in a real trading environment. The following five philosophies applied during testing phases can help unravel the root cause.

Backtesting Can Always Result in Perfect Scores

It's common knowledge that any algorithm can become the holy grail with the right combination of parameters when testing on historical data.

Figure 08 A Naive Perspective of Trading Algorithms



To understand this claim, an algorithm without any superior alpha, or insights, that runs on a full combination of parameters will return a standard distribution of profits, before all fees, taxes, and slippage costs. An algorithmic trader can then select the best parameters and start trading in the real environment. Unsurprisingly, most algorithms will fail due to the results of backtesting and the optimization phase.

The Real Trading Result Is the Only Thing That Matters

Testing on historical data is to form confident predictions on real trading performance after all. Suppose an algorithm predicts the result of a 20% annual return. What is the confidence level of this prediction in real trading? An algorithm is only ready to run in real life if a trader can expect a confidence level of at least 50%.

Correlation Is to Ensure Reliability

A high degree of correlation across each testing phase increases the reliability. On the contrary, a low degree indicates the testing phase has issues and the results may not be reliable. There are 4 main stages of algorithm evaluation:

- In-sample backtesting;
- Out-of-sample backtesting;
- Paper trading;
- Small account test.

The higher the correlation between these 4 stages, the more consistent and reliable the algorithm is in real trading.

Correlation	Evaluation
< 0.3	Not good
0.3 – 0.5	Acceptable
0.5 – 0.8	Good
> 0.8	Great

Out-of-Sample Backtesting May Not Be as Reliable

Many professional traders rely on out-of-sample backtesting after in-sample testing to evaluate an algorithm's reliability. Although this technique is widely accepted in machine learning, traders can apply it in the wrong way for algorithmic trading. The main reason is that out-of-sample data is still already a known fact traders may already be aware of.

For example, regardless of testing technique, all investors know that 2020 – 2021 is a strong bull market after the Covid-19 pandemic. And 2022 is a year of recession with high interest and high inflation. However, active traders may already be aware of this information before out-of-sample backtesting. Thus, it may not be reliable since out-of-sample data is not completely secret.

Adding Forward Testing to Ensure Objectivity

Forward testing on never-seen-before data ensures objectivity when evaluating algorithm performance. Note that paper trading is to objectively evaluate the performance. The small account test is to confirm both the technical implementation and the real performance altogether.

32

CRITICAL MISTAKES IN BACKTESTING

Testing on past data provides algorithmic traders a perspective into the future. They can form reasonable expectations on profit margins, maximum drawdown, and other key indicators. Some serious errors in this phase can make prediction results meaningless and invalid, and leave actual results to randomness.

Properly executing backtesting will form reasonable assumptions about the future trading results of the algorithm. Without careful design and proper assumptions, test results can create wrong expectations for high profits while consistently losing on real trades.

Below are five critical mistakes in the backtesting phase.

Overfitting

This is the most common mistake that gives algorithmic traders the illusion of finding the holy grail. Overfitting can appear in a variety of ways:

- Buy at the lowest price of the candle chart and vice versa. For example, buy a stock at 100,000 VND even though the current share price is 105,000 VND.
- Gain knowledge of future events or use future data to make decisions at the moment. A simple approach is to buy a stock if its price rises 20% next month. It can be done in backtesting and often gives outstanding results.

- Find the best set of parameters. By adding parameters to an algorithm, a trader can always find the best set that gives an outstanding return on investment. It's often confused between overfitting and optimization. Many experienced algorithmic traders still do not distinguish these two important concepts.

Excluding Transaction Fees

Many algorithmic traders set the transaction cost to zero, and expect a small difference in real trading. This is a mistake since transaction fees in the real market significantly reduce profit margins.

As of June 3, 2022, on the Vietnamese derivatives market, the total transaction fees in both directions are approximately 0.12% of trading value. Assuming an active algorithmic trader trades one long position and closes it every day, the investment result will be negative 30% at year-end by transaction fees only. If the expected portfolio return is at an impressive rate of, say, 25%, the trader will still lose at year-end.

The figure below simulates an algorithm with a 25% expected return before transaction fees. It shows the real return after fees in a year with two trades every day:

Figure 09 25% Expected Annual Return With Transaction Fees

Ignore Slippage

The next irrational assumption is zero price slippage. This assumption even causes more harm than zero transaction fees. The concept of slippage is the same as transaction fees. It occurs in every transaction, and the difference is the slippage may be almost twice the transaction fees. From hundreds of thousands of contracts traded on the Vietnamese derivatives market, ALGOTRADE expects a two-way slippage at 0.7 points or 0.25% of trading value with market orders. This cost can significantly reduce the profit margin and can turn a potential algorithm into a disastrous failure.

The figure below simulates an algorithm with a 25% annual expected return before any slippage. It shows the real return after slippage in a year with two trades every day using market orders.

Figure 10 25% Expected Annual Return With Price Slippage

No liquidity restrictions

This is an assumption that makes algorithms unable to trade. It's more common for large investors. The reason is that in backtesting phases, the algorithm can buy or sell an unlimited number of shares and contracts at the market price. In real trading, algorithmic traders may see positions unmatched in this scenario.

Fastest Speed on the Market

This is an assumption from many high-frequency traders. They often assume algorithms can achieve the best speeds in the current market. This assumption creates an illusion of outstanding performance but may in fact result in unmatched orders. Even worse, algorithms may only buy or sell at a matched price completely different from their expectations.

33

BACKTESTING MODULE

Backtesting is a core feature of third-party software like AmiBroker, TradingView, and MetaTrader. Below is a detailed analysis of the basic modules necessary to develop a system for backtesting.

Benefits of Backtesting Module

To institutional investors or professional algorithmic trading firms, backtesting is a critical screening phase to assess an algorithm's performance in the future.

When the number of algorithms to test increases, these tasks repeat and lead to a requirement for a standardized testing process. There's now a need to build standardized testing modules that specialize in testing historical data.

Developing algorithms on testing modules that simulate real trading at the same time allow reusability. Investors will work more efficiently on these modules without the need to program two separate versions.

Last but not least, testing modules are built in a simulated environment. They help limit errors caused by changing between different running environments. It also increases the similarity of test transactions to real trading ones.

Direction to Develop Backtesting Modules

Testing modules for an algorithm typically have three essential parts.

Simulate Securities Companies. This is a must-have to test trading algorithms on historical data in the long run. This part is to set up a simulated security company to receive orders, match orders, or even cancel orders from trading algorithms. Depending on the virtual account and real market conditions at the time of receiving a simulated order request, it decides to accept or reject the orders. After receiving new orders, it relies on market movements to match orders similar to real trading. A simulated security company has the basic features of a real company, without special cases such as partial order matching. Simulating securities companies requires lots of effort. However, in the long run, it helps investors test algorithms on historical data as objectively as possible.

Program Trading Algorithms. Trading algorithms can receive past data, return trading signals, and execute transactions with a simulated securities company to build a transaction history. The key point is the same piece of code can be applied in all stages: backtesting, forward testing, and real trading.

Report. A backtesting system needs to standardize test reports to facilitate horizontal development. The basic parameters that should be included are:

- Total asset value over time;
- Profit;
- Maximum drawdown;
- Sharpe ratio;
- Ratio of opening profitable/losing positions;

- Expected value when profitable;
- Expected value at loss;
- Longest profit streak;
- Longest losing streak;
- Drawdown statistics.

Transaction history is recommended to be exported in Excel format for easy access, inspection, editing, and illustration.

The logic of simulating securities companies and performance and risk assessment is the same for all algorithms. It should be developed separately for reuse in different algorithms. Investors can save effort and minimize programming errors. Developing these modules independently is to simulate the real market and to assess the performance so it will not be affected by any particular algorithm. The assessment will become as objective as possible.

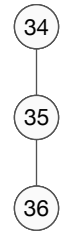
How to Build a Securities Company Simulator

Building a securities company simulator is a centerpiece of the backtesting module. In particular, it contains three main components:

- First is to build a paper trading broker that can receive orders, cancel orders, return order matches, and account status via API.
- Second is to determine whether an order can match, and in how much volume based on information obtained from market data (volume, price, buy, and sell) at the time of placing the order. The time will be in the past for testing past data, or in the present for testing future data. An example is from order matching: the simulator receives a limit order to buy at price X for ticker Y. If after receiving the order, and the order has not been canceled, the market price is less than or equal to X, then we can simulate that this order has been matched.

- Lastly, all information about placed orders, status, matching results, transaction costs, and account status at times should be saved in the database. These data will be used to assess the algorithm's performance and risk metrics.

In Vietnam market, note that the period of testing past data has lots of deviations from operations. In addition to common problems like taxes, fees, and slippage, traders should be aware of the volume and unobserved matches. The insufficient volume will reduce the algorithm's efficiency when operating in a real environment. Unobservable matching data will benefit algorithms using limit orders and negatively impact those using market orders.



CHAPTER V

OPTIMIZATION

34

OPTIMIZING TRADING ALGORITHMS

To trade in a real environment, a trading algorithm must have a logical foundation. It's different from patterns that generate profits but cannot be interpreted. These patterns can be found by exhaustively searching past data. Past data should only be used to apply financial theory to a particular market.

The optimization process begins with a hypothesis in the form of a preliminary algorithm. It can be proposed by a professional based on their financial knowledge or practical experience. This preliminary algorithm is optimized on a sufficiently long historical data range to produce a complete, stable, and profitable algorithm. The past data range is called the in-sample data of the fine-tuned algorithm. This data range needs to reflect market volatility. The algorithm needs to make sufficient trading orders in this range for its performance to be statistically significant.

A preliminary algorithm is refined by adding new rules and adjusting parameters so that its efficiency in the target market is optimal. The efficiency is measured by an objective function, and the target market is represented by an in-sample data set.

The optimal efficiency is the right balance between risk and return, and not when the algorithm reaches the highest return. The Sharpe ratio is often selected as the objective function. Limiting the risk of loss also helps an algorithm avoid overfitting. It's a key issue in the algorithm development process.

Overfitting is a special phenomenon when an algorithm is specifically fine-tuned to get optimal results in specific sample data. However, when the tuned algorithm works on out-of-sample data, the efficiency is significantly reduced. When searching for the optimal parameters, algorithm developers should note two requirements:

- Good performance in in-sample data;
- Equivalent performance in out-of-sample data and in real trading.

It's key to limit overfitting when developing the trading algorithm. Even though there are solid techniques in this area, it's challenging to remove overfitting completely.

Find an Optimally Fine-Tuned Algorithm

There are two steps to finding an optimally fine-tuned algorithm.

1. Add the Variational Law

When optimizing an algorithm, the dynamic nature of the market should be noted. Instead of choosing fixed parameters, it's advisable to create a rule that is a function of the statistical market variation. In this method, trading algorithms can adjust to the moving market, ensuring stability regardless of market conditions.

For example, volatility is an important indicator of market conditions. An algorithm may work differently in different market volatility. The average difference between the highest and lowest price of VN30 in recent days can be a statistical signal for market volatility. Using this signal, an algorithm can adapt itself to work well in different market variations.

2. Parameter Search

After adding market statistics to algorithms, there are possibly algorithm parameters that don't change frequently and need to

determine their values. Parameter space is the set of all possible combinations of these parameter values. The optimization process should be designed to pick the optimal parameter values.

A simple approach is that for each set of parameters, algorithm developers run backtests over past data and pick the best set according to an objective function. This function can be the Sharpe ratio or another criterion that balances risk and return. This method has the limitation that the number of parameter sets can be infinite or very large. In this scenario, there will be insufficient resources to try all possible sets. Another issue is that the set found in this method may not guarantee similar performance in future data. These are the two main problems in the optimization process.

Other methods have been proposed to limit the search space and still select a near-optimal parameter set, such as grid search, priority parameter search, and advanced search algorithm.

Grid search. This method is closest to the naive method above. All preset values of the parameter set will be checked. To avoid overfitting, the appropriate set of parameters will be reselected by dividing the data set in the sample into training and validation sets.

For example, a regression algorithm predicts how much the closing price of VN30F1M will change compared to the previous day's. It then opens and closes a position at ATO and ATC respectively. For instance, if it predicts a 3% decrease, it will open a short position at ATO and close at ATC. It will make a profit if the prediction is correct. In this algorithm, there are two main parameters to identify:

- N: the number of previous days used as data for prediction.
- Alpha (α): the threshold to open a position at ATO. If the price difference is greater than this threshold, the algorithm will open a position at ATO.

In this example, there are 25 sample values of N, and 7 sample values of α to check. Also, the parameter search assumes this range is sufficient to cover the parameter space that is likely to contain the optimal value. Therefore, there are 25 x 7 = 175 different parameter sets. Tables 01 and 02 list the values and performance of these parameters. Based on the following result, the optimal parameter is N = 19 and α = 0.2%. It results in 127.62% profit and a -11.83% maximum drawdown (MDD).

Priority parameter search. When there are too many combinations of parameters to evaluate, grid search will be inefficient due to a lack of computaional resources. To save time, the parameters that have a great influence on the algorithm should be prioritized by only changing the value of that parameter, while keeping others unchanged.

We can fix α = 1.0%, and change the values of N to the optimal N which is 15. Then we fix this parameter and search for different values of α. Finally, we find the optimal α = 0.3% (and N = 15) with 103.95% profit and -16.20% maximum drawdown (MDD). We find this parameter set by trying 31 different combinations, instead of 175 as with the grid search method.

Advanced search algorithm. In advanced search, it's recommended to discover which direction we should move in the parameter space to find the optimal set, guided by the value of the objective function. In this way, it's only required to test a much smaller part of the parameter space to find the optimal approximation parameter set.

One of the advanced methods is the hill climbing search. This method searches based on a direction that gradually increases efficiency. It stops when there are no other parameter sets around for higher results. For example, the direction in Table 01 is [N = 21, α = 0.3%] → [N = 20, α = 0.3%] → [N = 19, α = 0.3%] → [N = 19, α = 0.2%].

Table 01 Rate of Return of Mean-Reversion Algorithms on In-Sample Data

	Alpha (α)						
N	**1.00**	**0.50**	**0.40**	**0.30**	**0.20**	**0.10**	**0.00**
10	31.14	31.80	36.25	39.21	28.89	23.35	40.49
11	56.37	44.78	40.92	57.59	53.29	44.98	27.81
12	40.28	30.04	40.70	32.61	57.54	62.84	63.26
13	61.11	53.82	69.78	72.87	92.04	95.15	101.91
14	65.49	60.00	61.07	67.03	108.66	108.37	109.46
15	65.87	87.48	91.35	103.95	105.67	96.87	106.08
16	44.41	73.34	74.29	100.07	96.99	78.48	66.47
17	44.38	66.81	97.42	91.91	99.71	93.14	88.48
18	33.88	80.73	89.67	83.56	114.37	103.67	106.45
19	23.05	56.39	86.08	94.80	127.62	106.80	77.59
20	26.33	79.93	78.89	89.22	97.79	106.73	119.42
21	28.73	62.44	74.08	81.40	94.34	120.04	126.37
22	26.62	65.72	73.89	83.91	100.20	115.15	106.74
23	23.83	70.45	80.64	80.30	79.38	105.56	94.19
24	23.81	64.85	85.31	78.46	87.39	103.04	105.60
25	15.67	70.25	84.69	77.35	89.31	119.43	120.79
30	12.27	48.63	84.17	71.70	83.77	128.11	120.72
35	25.69	49.60	59.57	54.42	60.35	90.48	70.46
40	14.46	48.11	41.14	34.14	42.18	60.89	67.19
50	21.73	31.15	25.46	31.83	22.40	37.99	32.35
60	2.74	13.87	27.10	8.51	3.29	16.87	18.50
70	0.94	17.40	2.87	16.82	8.12	23.82	47.46
80	1.78	10.61	1.29	9.50	1.45	27.76	24.63
90	6.12	27.96	23.99	21.63	17.79	13.21	36.59
100	9.16	25.68	15.70	22.87	14.39	28.82	28.12

Table 02 MDD of Mean-Reversion Algorithms on In-Sample Data

	Alpha (α)						
N	**1.00**	**0.50**	**0.40**	**0.30**	**0.20**	**0.10**	**0.00**
10	-14.86	-25.02	-26.00	-31.51	-31.13	-30.31	-34.20
11	-10.56	-18.25	-22.49	-22.94	-18.25	-20.79	-20.24
12	-14.40	-16.77	-20.80	-24.19	-31.17	-28.79	-25.55
13	-10.56	-13.61	-16.00	-18.03	-18.75	-20.48	-20.36
14	-10.56	-11.89	-13.28	-19.43	-18.86	-18.20	-19.50
15	-11.02	-11.96	-17.24	-16.20	-21.16	-28.80	-35.84
16	-16.18	-14.07	-14.57	-14.23	-16.08	-22.01	-26.17
17	-16.10	-16.10	-16.10	-12.74	-13.29	-23.04	-21.59
18	-15.92	-14.88	-14.88	-14.98	-14.73	-17.72	-18.86
19	-21.27	-14.25	-15.03	-12.33	-11.83	-18.58	-21.28
20	-16.75	-12.78	-15.02	-14.22	-16.73	-16.50	-18.74
21	-15.14	-10.56	-11.11	-15.18	-17.51	-19.10	-19.01
22	-16.69	-12.61	-11.35	-11.58	-13.52	-17.64	-19.11
23	-17.79	-10.56	-12.89	-11.49	-13.88	-16.95	-27.26
24	-17.43	-10.56	-10.56	-12.53	-11.81	-21.59	-29.77
25	-18.05	-10.57	-12.69	-12.45	-13.52	-17.79	-24.13
30	-10.58	-13.61	-10.56	-12.58	-16.58	-14.04	-16.36
35	-13.66	-12.88	-12.68	-11.69	-15.37	-15.68	-20.98
40	-14.63	-13.76	-15.57	-23.86	-21.67	-26.50	-21.43
50	-12.62	-29.78	-29.26	-27.91	-23.62	-20.80	-24.83
60	-14.67	-27.78	-27.35	-33.80	-29.71	-36.81	-34.31
70	-23.09	-23.50	-32.18	-36.76	-49.66	-34.92	-36.13
80	-18.91	-24.01	-30.59	-30.11	-35.91	-28.74	-45.10
90	-17.91	-25.29	-23.65	-34.18	-41.32	-30.29	-38.98
100	-14.21	-20.97	-32.35	-31.40	-39.82	-32.63	-45.89

35

TECHNIQUES TO AVOID OVERFITTING

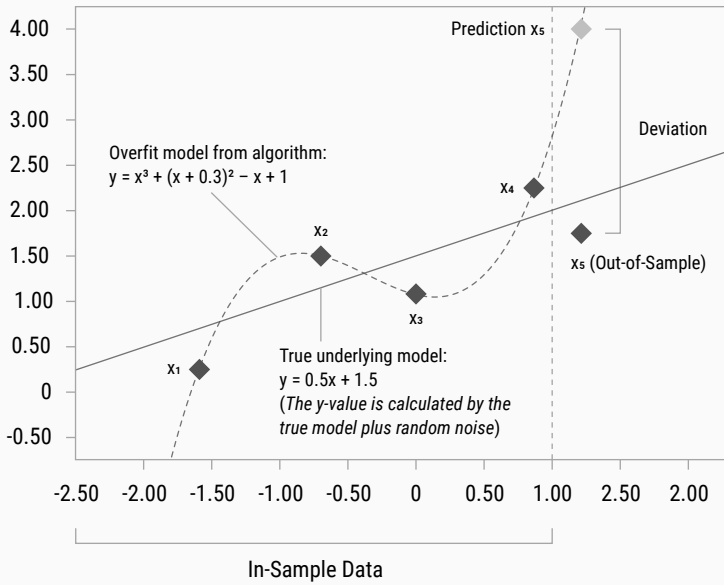
Overfitting Definition

In statistics, overfitting is the result of an analysis that is too precise on a particular data set. It fails to perform well in another data set or to predict future data. This phenomenon occurs when the data noise, or residual/unexplained variation, is included to make an unnecessarily complex model, while it should instead be explained by randomness. Since the complex pattern is influenced by random factors, it will not hold true in the future.

For algorithmic trading, overfitting occurs because a fine-tuned algorithm uses models and parameter values that increase the test performance in the sample by randomness. That randomness is unlikely to repeat in the future, which leads to incorrect predictions.

The following figure shows an in-sample overfiting (cubic curve) in a regression problem.

Figure 11 Example of Overfitting Data



For example, in algorithmic trading, we can have many different versions of the same algorithm. Each has better performance on the in-sample data compared to the previous version because of added rules. The later version gets increasingly more complex than the former ones. The performance of these versions is in the table below. The first two versions of the algorithm perform well with positive return on out-of-sample data despite the acceptable difference in performance compared to the in-sample data. Meanwhile, version 3 adds new rules and increases in-sample performance, while significantly reducing out-of-sample performance when compared to version 2. In this case, we can conclude that version 3 is an overfit, and version 1 is an underfit since it can be improved to outperform in both in-sample and out-of-sample data.

Table 03

	In-Sample	Out-of-Sample
Version 1		
Monthly profit	2.03%	1.07%
Average profit per trade	0.151%	0.316%
Sharpe ratio	0.86	0.52
Version 2		
Monthly profit	2.72%	1.67%
Average profit per trade	0.147%	0.133%
Sharpe ratio	1.11	0.54
Version 3		
Monthly profit	3.38%	0.69%
Average profit per trade	0.121%	0.033%
Sharpe ratio	1.09	0.16

Techniques to Avoid Overfitting

Here are 05 techniques to avoid overfitting. It should be noted that the techniques only help to eliminate basic overfitting, not undetectable cases. For the latter, we need to use the algorithm testing process after optimization.

Understand Why a Rule Is Right

When finding a rule that has produced profits in the past, it's important to explain why the rule produces such returns based on financial theory, or human behavior, not hoping the history will repeat itself in the future. This makes the rule valuable. Combining financial knowledge and market conditions, investors can properly interpret and assess the algorithm's performance when monitoring the trading system. As a result, algorithmic traders can predict the irrationality the moment when this assumption no longer holds true and promptly stop the system or change the algorithm before any disaster.

Split the Data Into Training and Validation Sets

After splitting the data set, we only use the training set to optimize the algorithm's performance. The validation set is to find a version on the optimization path with a reasonable fit to ensure equivalent performance on both data sets (Figure 12).

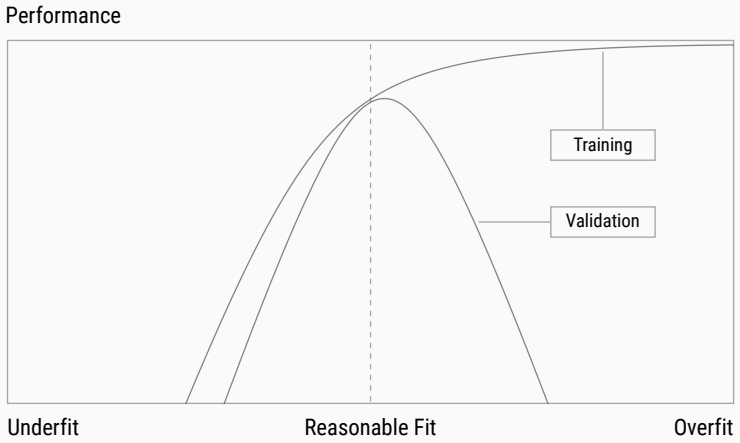
The idea behind is that randomness in the training and validation sets are different, but the two share a common hidden pattern. Therefore, if a pattern consists of too much randomness only in the training set, it will perform poorly in the validation set where the randomness does not occur. If a version performs well on both data sets, it's likely the hidden patterm has been found, and the correct rules and parameters have been identified.

In Table 03, using out-of-sample data as a validation set, we can select version 2 as the reasonable fit.

Choose a Simple Preliminary Algorithm to Start With

When an expert proposes a preliminary algorithm to test on the target market, there will be a set of multiple versions that can be applied. These versions are basic algorithms but have different enhancement rules and parameter values. During the

Figure 12 Algorithm Performance in the Optimization Process



optimization process, these versions are tested to find the most likely version that can perform well in the real market. If the preliminary algorithm is simple enough, these versions will have a lesser chance of overfitting.

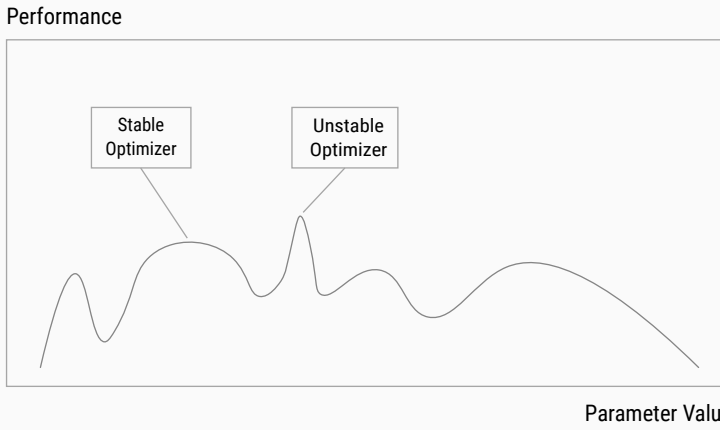
In figure 11, we see the linear model $y = 0.5x + 1.5$ is simpler then the cubic polynomial $y = x^3 + (x + 0.3)^2 - x + 1$. The hidden patterns are mostly simple in real life.

Drop the Preliminary Algorithm if There Are Too Few Versions for Profit
After evaluating different versions of the initial algorithm, if there are too few instances that yield profits, the algorithm hypothesis may not be valid for the target market. The few versions with positive performance may be due to randomness or overfitting of the training data. We should drop the preliminary algorithm at this stage.

Better to Choose a Stable Optimal Point Than a Sharp Optimal Point

Financial markets are always volatile. The optimal parameter value may change in the future. If a sharp, unstable optimal point is selected, it's likely the performance will decrease rapidly even though the optimal parameter value changes ever so slightly.

Figure 13 Example of Optimizers



36

POST-OPTIMIZATION ASSESSMENT

An algorithm is considered effective at a point in time if it remains profitable as intended when applied in a target market in the future. After the optimization process, we likely have an effective algorithm on the in-sample data set. However, this algorithm is fine-tuned on the same data set, and there's a possibility of overfitting. To determine the effectiveness of an algorithm, we need to assess the algorithm on never-before-seen data that has never been used to build the model or to fine-tune its parameters. This data set is also called out-of-sample data. Note that this data should be sufficiently representative for the performance statistics to be valid.

Testing the Algorithm on Out-Of-Sample Data

To assess a trading algorithm, we need to carefully evaluate its performance on an out-of-sample data set to find and explain performance dissimilarities if any when applied to in-sample data. The assessment also helps understand the advantages, risks, and behavior of the algorithm in different market conditions.

This is key to monitor the algorithm during real trading. From this point, we can detect anomalies to manage risks and find opportunities to improve the algorithm.

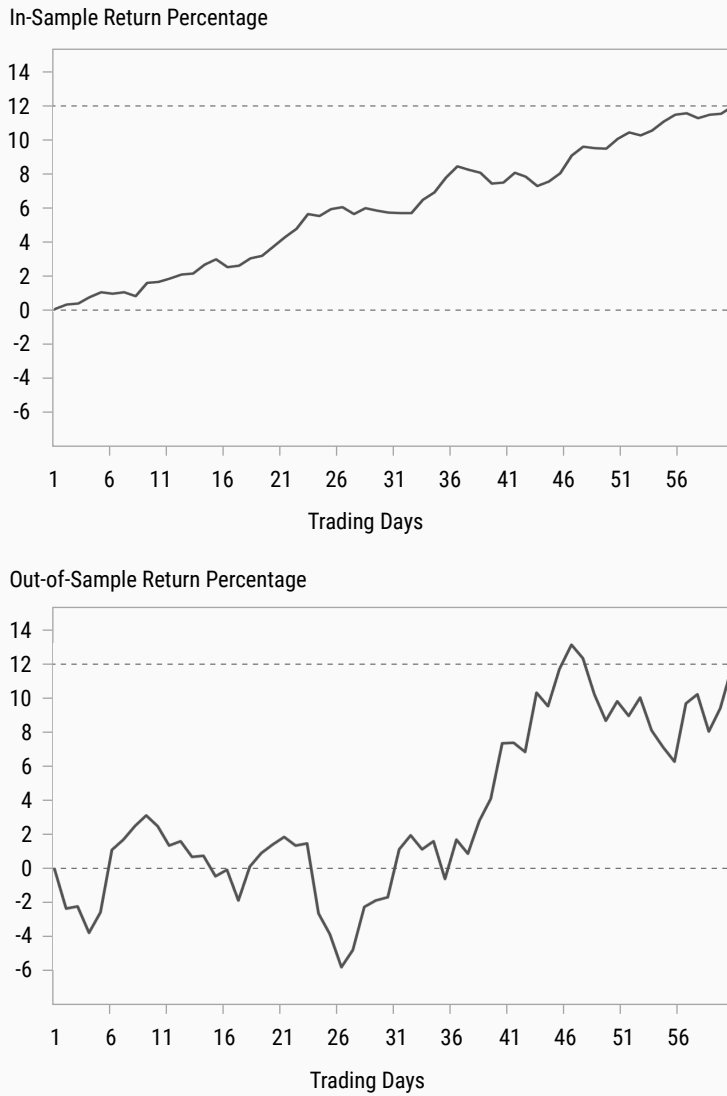
The effectiveness of a trading algorithm is evaluated by a comprehensive analysis of transaction records. They include net assets over time and statistics on buys, sells, wins, losses, and the times when the net asset draws down. In these metrics, the maximum drawdown is to assess risk, while monthly returns are used to evaluate rewards. The net asset curve provides an overview of both. The detailed transaction statistics report the algorithm's behaviors.

An algorithm is considered ineffective if its performance on out-of-sample data shows abnormal results compared to in-sample data. Some examples are losing streaks for a long time, sharp drops, excessive deviation of daily returns, or unmatched risk and reward compared to in-sample performance. Investors should carefully research and identify the reasons for these anomalies to determine the algorithm's effectiveness. For example, it's advisable to pay attention to market conditions in the data. It's a possibility of performance mismatch due to differences in market conditions, not because of the algorithm itself.

Figure 14 is a real-world example of the performance dissimilarity of risk and reward performance between in-sample and out-sample data.

A common way to detect the incompatibility of two trading profiles is to use maximum drawdown (MDD) as a metric. The algorithm will be considered ineffective and unusable when it undergoes a sharp drop exceeding a predefined multiple of in-sample MDD. It's often used to prevent serious losses when the algorithm is applied on the main account with large capital.

Figure 14 Comparison Between In-Sample and Out-Of-Sample Performance



Algorithm Testing Process

The testing process for a trading algorithm has 4 main stages.

Testing on Historical Out-Of-Sample Data

In this first stage, we test the algorithm on historical out-of-sample data. If its result shows potential and similarity to in-sample performance, we can move to the paper trading phase.

Paper Trading

In this stage, we test the algorithm on future data. That is, the algorithm is simulated to operate in a real-time trading environment. When receiving a transaction order, algorithm developers will not know in advance how it affects the end result. The goal is to ensure no one can predict the performance of paper trading as it operates in real-time.

The main difference between the first two stages is that the historical out-of-sample data may be used repeatedly for several algorithms. It leads to the possibility of an algorithm overfitting this data set, though not trained and validated on it. This phenomenon is called data depletion. Paper trading helps avoid this since it uses completely new data in the system.

If the algorithm still performs well in paper trading, we know that the algorithm is not an overfit and can proceed to the next phase.

Small-Account Test

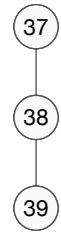
Paper trading simulates the process from placing an order to its completion. This simulation may not be completely accurate due to data limitations in the Vietnam market. For example, it's not possible to accurately simulate partial order matches in the derivatives market. The small-account test is designed to solve this problem. It tests the algorithm on future data using a real account with small capital.

Although the performance difference between paper trading and small-account tests is unavoidable, the difference really depends on the algorithm, assuming we provide the best simulation of a security company in paper trading using market data. The goal of a small-account test is to quantify this difference.

If the performance does not result in noticeable deviations, the algorithm can proceed to the final stage to make profits for investors. We move on to real trading on the main account with large capital, where close monitoring is necessary when first applied to the target market.

Live Monitoring During Real Trading

At this stage, investors can be confident that the algorithm is not an overfit. Yet the algorithm can become ineffective when factors affecting stock prices have changed. The algorithm may need regular retesting in real trading. When the algorithm's assumption is no longer valid, the algorithm needs to be stopped or adjusted accordingly before causing catastrophic losses.



CHAPTER VI

FORWARD TESTING

37

MEANING OF FORWARD TESTING

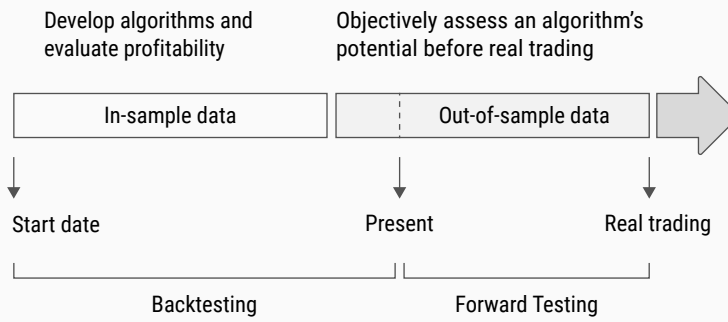
When coming up with a trading idea or a trading strategy, algorithmic traders will model their idea into a specific algorithm. They then use historical data to test whether an algorithm is profitable or not. They evaluate how this profitability will likely continue in the future to make investment decisions.

In a real trading environment, however, the target market often changes. This can lead to many different scenarios. Some algorithms are only profitable when the market trends upward. Some yield profits in trending markets but undergo big losses when the market does not show any particular trends.

In addition, it's possible the random test data is perfect for the algorithm, leading to highly positive test results. In practice, it may be difficult to repeat the same scenario. Therefore, the process of developing and evaluating an algorithm should try to test many different scenarios to objectively assess the algorithm's potential before real trading.

Figure 15 illustrates the process that combines past and future data in the algorithm development and evaluation process.

Figure 15 Distinguish Between In-Sample and Out-of-Sample Data



- **In-sample data**: a part of past data, also known as in-sample past data, to test and optimize the algorithm. Typically 70% of past data will be used for this purpose.
- **Out-of-sample data**: includes out-of-sample past data and future unseen data. The purpose is to test an algorithm after the optimization phase whether it can generate stable profits. Out-of-sample data and in-sample data should have no overlap.

For example, it's necessary to test algorithms with the recent 10-year price and volume data from 2012 to 2022. The data from 2012 to 2019 will be the in-sample past data and the rest is out-of-sample past data.

The backtesting process uses both in-sample and out-of-sample data. Normally this data set is sufficient to evaluate algorithms. However, since the testing only uses past data, there's a possibility of common errors such as skipping slippage and overfitting.

To avoid common errors, the evaluation process takes an extra step to test future data. Forward testing, known as real-time

algorithm testing, uses out-of-sample data that is not available at present. Its purpose is to add objectivity to the evaluation process while evaluating the algorithm's feasibility.

Forward testing has two main stages. The first is paper trading, which is similar to real trading but no actual transactions are made. All transactions are simply recorded for further evaluation. The second stage is performed in a real environment with a small amount of capital. It is to finally test the effectiveness of the algorithm, its hypothesis, and any technical issues in the system. This stage is thus called a small account test.

Backtesting and forward testing are both essential in the algorithm development process. Neither is better than the other. High profits and similar results in both types of tests are important criteria to decide whether to use a particular trading algorithm.

38

PAPER TRADING

Paper trading is the first stage of forward testing. Future data is unseen data. Paper trading thus eliminates overfit from backtesting and the parameter optimization phase. In addition, all transactions in this stage are at the present. We can observe each closely and see the context clearly. It's easier to deduce and explain the algorithm's behaviors and its efficiency, thereby improving its performance. In practice, this stage typically lasts 2 months. Below are key points of paper trading to keep in mind:

- *Ensure consistency*. Do not update, or change the rules, or any algorithm parameters. Any small change in this phase can completely lose the meaning of the testing process: the future data set then becomes equivalent to the past data set. Paper trading becomes the optimization phase. In the necessary event that a new rule or parameter adjustment is required, one should consider restarting the entire testing process.
- *Ensure real-time*. Simulation in paper trading should be closest to real environments. The only difference is the order matching results returned from securities companies and any price slippage incurred. Taxes and fees should be fully calculated. Ensuring real-time records help algorithmic traders have an accurate, intuitive view of the algorithm. If any transaction is delayed compared to real-time, the difference between forward testing and backtesting will no longer be meaningful.

- *Build test criteria prior to starting*. The algorithm is tested to assess its ability to generate stable profits in the future. Paper trading also assesses the similarity between backtesting and forward-testing results. All the criteria need to be defined so the algorithm's performance can be verified upon completion.

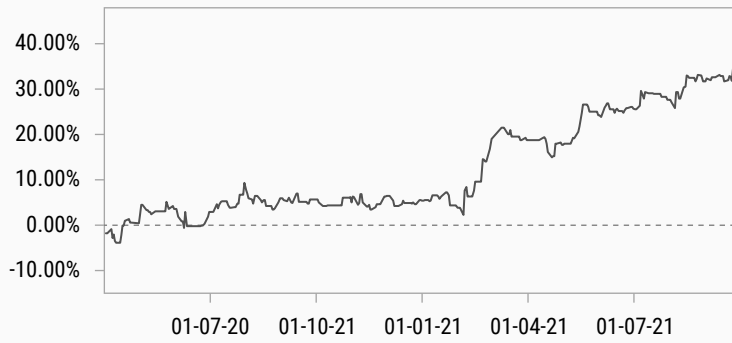
Figure 16 illustrates how an algorithm is evaluated in the paper trading phase.

Figure 16 Paper Trading to Evaluate an Algorithm

Out-of-Sample Backtesting Performance[1]:

Trading Period	Profit	MDD	Sharpe Ratio
09/04/2020 - 27/08/2021	34.05% (1.95% monthly)	-6.98%	4.88

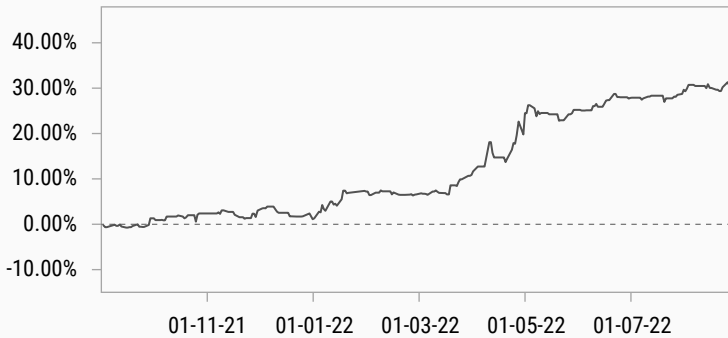
Accumulated Return Chart:



Paper Trading Performance:

Trading Period	Profit	MDD	Sharpe Ratio
30/08/2021 - 26/09/2022	31.25% (2.33% monthly)	-4.38%	7.13

Accumulated Return Chart:



Leverage in a Real Environment: 3.5

Reasonable Expectation in a Real Environment:

Monthly Return	Annual Return	MDD	Sharpe Ratio
3.50%	51.10%	-36.65%	1.40

Stopping Criterion: MDD reaches 50% or annual return under 7%.

Conclusion: The algorithm passes the paper trading test.

---

[1] We assume out-of-sample backtesting and paper trading did not use leverage during testing. In a real environment, we use 3.5 leverage.

At the end of paper trading, the algorithmic trader can test the similarity between the trading performance of the out-of-sample and in-sample data, besides the algorithm's profitability. At the same time, they can estimate the risk and performance of the algorithm as well as reasonable expectations when the algorithm operates in real life. They can also calculate the Kelly criterion and the algorithm's compatibility with the current system to make the final decision.

Note that a good trading algorithm still has bad time periods when it cannot be profitable. As long as the algorithm behaves similarly for past periods with similar current market conditions, its future profit potential should still be considered.

39

SMALL ACCOUNT TEST

Before the algorithm goes live, trading on a small account is part of the forward testing phase closest to real trading. This phase has the following unique benefits.

Hypothesis Testing

All assumptions in the previous testing stages like price slippage, algorithm activations per day, and order matching rate will be revealed in the real environment. However, some specific assumptions can only be tested with large capital accounts and not at this stage.

Test the Trading System

All other testing phases do not involve the automated trading system directly. This is the phase to fix any remaining bugs and to check how well the algorithm works with the operating infrastructure and the algorithmic trading system. The trading system is considered to be working properly when the operating algorithm is 99% stable compared to the ideal assumption. In principle, a small account test should match the results of paper trading. The following are some key differences to consider when evaluating algorithm performance at this stage.

- *Order matching time*. The previous tests assume that the orders are executed as soon as a signal appears. They do not account for price delays, data processing, and order placements.

Comparing order matching time to ensure low deviation (less than 02 seconds) is a very important criterion at this stage. For some strategies that require high speed, code execution optimization can improve performance. It reduces execution latency and the possibility of price slippage, maximizing performance and taking more opportunities.

- *Order execution price*. The execution price is usually assumed at the current price minus slippage. If calculated correctly, this assumption will be reasonable in the long run. If there are more than 20% trades with matched prices different from the expected 0.3 points in derivatives market, traders should verify their assumptions.
- *Partial or no match*. This case is often overlooked in a trading simulation. However, it's inevitable in a real trading environment with a sufficiently large account.
- *Matching ratio vs assumption*. Compare the order matching statistics in small account tests with previous tests. It's especially important for algorithms using limit orders.
- *Operating time (uptime)*. Check system stability. If the system works under 99% of the trading time, it should not proceed to real trading.

Keep in mind that, for trading multiple algorithms, the trading system has its own testing method. A small account test is only to check how well the algorithm fits with the existing system. It's not a test for the trading system itself.

Prevent Overfitting in Sufficiently Long Operations

This is the most crucial point in small account tests: any overfit algorithm will likely fail at this stage. Typically 30 transactions are sufficient to evaluate the price momentum strategy.

To determine whether poor outcomes of small account tests stem from randomness or faulty assumptions, the algorithmic trader must analyze the situation carefully. More than 95% of the time, the problem lies in making incorrect hypotheses.

Figure 17 shows how to use an algorithmic benchmark to evaluate a small account test.

Figure 17 Evaluate Algorithms in Small Account Tests

Passing Criterion 1 (Stable Expectation During Operation):

Monthly Return	Annual Return	MDD	Sharpe Ratio
…	…	…	…

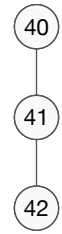
Passing Criterion 2 (General Performance):

Annual return	MDD
…	…

Performance in Small Account Test:

Trading period	Return	MDD	Sharpe Ratio
…	… (… monthly)	…	…

If the test meets either of the criteria above, we can proceed to real trading.



CHAPTER VII

REAL TRADING

40

ALGORITHM OPERATION IN REAL ENVIRONMENT

After completing 08 stages from hypothesizing to small account tests, the algorithm can now trade on the real market and earn a profit. Handling 100% of transactions automatically is a step towards "financial independence", "financial freedom", or "passive income". However, there are still a few critical aspects to consider in this phase.

Parameter Setting

The algorithm parameters for real market trading is usually the same as for small account tests, except for capital size. However, this may not be the same with hyper-large capital. It's because investors have to modify the parameters during operation to ensure the algorithm's reliability in the real market versus the testing phase. In particular, the market-making and market-neutral strategies require special attention.

Define Expectations

Based on backtesting, forward testing, and the algorithm strategy, traders can set basic expectations for the algorithm to detect any unexpectedly large deviations. Note that the criteria differ for each algorithm. However, here are some common criteria to meet before running the algorithm in real trading:

- Expected annual profit;
- MDD;
- Sharpe ratio;
- Average slippage.

Some algorithms need leading indicators and these indicators are often more important than the ones above. Some leading indicators for limit order strategies are as follows.

- Number of signals per week;
- Number of matched orders/signals.

Traders need to monitor any significant difference in any of the criteria and learn to adapt to this new information, especially during the first 03 months of algorithm operation. They also need to prepare for the worst-case scenario to stop the algorithm at a predefined threshold, usually a multiple of MDD.

Online Monitoring

At the start of the live operation, traders should monitor online to handle any real-time issues. Such can come from data, algorithms, systems, securities companies, and reporting systems. These issues are especially frequent with the first algorithms. In some special cases, the system cannot close the opened position, and manual intervention is required. After acquiring enough experience, algorithmic traders can skip this task.

Fix Technical Bugs

Even after many testing phases, an operating algorithm is still not bug-free. It's totally normal to have occasional errors. However, algorithmic traders should fix bugs as soon as possible. In our experience, technical failures often happen in the first month, and then drop by 90% in the next months.

System Fine-Tuning

There are three common factors that lead to system tuning:

- *Financial markets are always changing*. Some changes are quick and sudden, such as laws and policies. This can completely change the investment environment and it's necessary to fine-tune the system to quickly adapt.
- *The original condition to form the algorithm hypothesis has changed*. Over time, more data and information are updated, making the initial hypothesis no longer valid. As these conditions change, traders should consider testing for plausibility and further fine-tuning. If the change is too massive, algorithm traders should consider forming a new algorithm instead.
- *Real trading data provide further important information*. This is an effective feedback loop in algorithmic trading. Traders create algorithms and algorithms generate data. This gives algorithmic traders more information and more relevant perspectives. They can upgrade the algorithm to a more complete version.

Capital Allocation

It's important how the trader will act when the algorithm makes a profit or a loss. Will they use the strategy of having a fixed capital for the algorithm, meaning withdrawing profits if any? Will they choose a compound interest strategy that will use the maximum available capital at each time? This decision can greatly affect the long-term performance of algorithmic trading, especially for multi-algorithm trading.

After backtesting and forward testing in a scientific way, algorithmic traders can trust the profitability of their algorithms, at least in the short term without major changes in the environment. They still, however, need to check for risks periodically.

41

EVALUATION OF EXECUTION ALGORITHMS WITH TWAP AND VWAP

Trade execution evaluation measures and compares how well different execution algorithms work. So traders can choose the best ones to achieve their trading goals and reduce transaction costs.

Execution efficiency is measured by comparing the average order execution price with a reference price. It's to see if the execution price was too high or too low. Two common reference prices are: volume-weighted average price (VWAP) and time-weighted average price (TWAP).

Volume-Weighted Average Price

VWAP is the average price of all trades executed during the calculation period, weighted by their volume. It's calculated using the following formula:

$$VWAP = \frac{\sum P_i Q_i}{\sum Q_i}$$

Where:

- P_i is the trade execution price for trade i;
- Q_i is the execution volume of trade i.

VWAP reflects all market activity, supply and demand for all market participants, so it provides a reasonable standard for evaluating trade execution.

Here's the formula to evaluate trade performance:

$$VWAP\ Cost = S \times \frac{\overline{P} - VWAP}{VWAP}$$

Where:

- $\overline{P}$ is the average order execution price;
- S is the direction of the opened position (S = 1 : long position, S = -1: short position).

For example, an investor executes a buy order with an average order execution price of 20,500 VND. VWAP during the trading day is 20,000 VND.

$$VWAP\ Cost = 1 \times \frac{20.500 - 20.000}{20.000} = 2.5\%$$

This means that investors bought on average 2.5% higher than other buyers during the day.

Time-Weighted Average Price

TWAP is the simple average price of all trades executed during the calculation period:

$$TWAP = \frac{\sum P_i}{N}$$

Where:

- P_i is the execution price for trade i;
- N is the number of execution price.

TWAP does not consider trading volume. So investors use

TWAP when they want to ignore outlier trades. Outlier trades can be caused by large orders at low or high prices during the calculation period.

Here's the formula to evaluate execution performance:

$$TWAP\ Cost = S \times \frac{\overline{P} - TWAP}{TWAP}$$

Where:

- $\overline{P}$ is the average order execution price;
- S is the direction of the opened position (S = 1: long position, S = -1: short position).

Trading Application

For index investing strategies, especially for mutual funds, the annual returns are usually similar to the benchmark index. In this case, transaction execution may be the most important factor in the end result besides administrative costs, fees, and taxes.

The following is an example of how execution costs affect two index funds:

Investment Fund	A	B
Paper return	9.00%	9.00%
Management fee	- 0.30%	- 0.30%
Trading fee and taxes	- 0.50%	- 0.50%
Execution cost	- 2.00%	- 0.00%
Actual Return	**6.20%**	**8.20%**

As shown above, investment fund B performs 2% better than investment fund A. In this example, the only difference in performance is the main reason for the success or failure of the fund manager.

Execution costs also matter for individual investors in Vietnam. Frequent trading and the high volatility of each stock can make individual investors overlook the impact of execution costs. However, with an account turnover rate of 2 times per month, i.e. 24 times per year, an average execution cost of 0.5% is equivalent to 12% annually. In a normal market, it would not be surprising if 95% of individual investors lose money with this execution cost.

42

IMPLEMENTATION SHORTFALL

To test if the trading algorithm works well in real trading, we need to compare the paper trading results and the actual trading results. Algorithmic traders can start by measuring and analyzing the reasons for any differences. They should fix problems that might happen during trade execution. A commonly used analytical method is to use implementation shortfall.

Formula

The implementation shortfall (IS) measures how much the paper profit differs from the actual return. We can use the following formula:

$$IS = Paper\ return\ - Actual\ return$$

Paper return is the theoretical profit if we could fill all our orders at the desired price and volume without incurring any costs. The formula is:

$$Paper\ return = (P_n - P_d)(S) = (S)(P_n) - (S)(P_d)$$

Where:

- S is the total volume of open positions, S > 0 in the case of opening a long position, while S < 0 for a short position;
- P_d is the desired price when we decide to open a position;
- P_n is the current price.

The actual return is what we really earned based on executed trades. We can use this formula:

$$Actual\ return = (\sum s_j)(P_n) - \sum s_j p_j - Fees$$

Where:
- s_j is the matched volume in the transaction j;
- p_j is the matched price in transaction j;
- Fees are transaction fees and applicable taxes.

Combining the above formulas, we get:

$$IS = \sum s_j p_j - (\sum s_j)P_d + (S - \sum s_j)(P_n - P_d) + Fees$$

IS is decomposed into three components, explaining the three reasons for the discrepancy:

- *Price slippage*: the price may change and may not match our plan from the decision-making moment to the time of placing an order;
- *Opportunity cost*: matching volume is not enough compared to our plan;
- *Fees*: transaction fees and taxes.

Price slippage can be further decomposed into two more components: delay costs and transaction costs.

$$IS = (\sum s_j)P_0 - (\sum s_j)P_d + \sum s_j p_j - (\sum s_j)P_0 + (S - \sum s_j)(P_n - P_d) + Fees$$

Where P_0 is the stock price at the time of placing the order.

- *Delay cost*: when an order cannot be placed immediately after the decision is made, it results in a price change from the decision-making moment to the time of order placing;

- *Transaction cost*: depending on market liquidity, order type (limit order, market order, etc.), or system errors may result in orders being matched at different prices.

Example of Implementation Shortfall

Assume that at 09:15, the share price of VINGROUP (HOSE: VIC) is 60,000 VND per share, and we decide to buy 10,000 VIC shares.

At 09:20, the system starts to place orders. At this moment, the stock price has gone up to 60,100 VND.

At the end of the day, the closing price of VIC is 60,800 VND.

By the end of the trading day, we can only buy 8,000 shares and pay an average fee and tax of 200 VND per share.

Timeline	Price	Volume
Investment decision	$P_d = 60{,}000$	$S = 10{,}000$
Place order	$P_0 = 60{,}100$	$S = 10{,}000$
Order matched	p_1, p_2, p_3, p_4	s_1, s_2, s_3, s_4
Close position	$P_n = 60{,}800$	$\sum s_j = 8{,}000$

Below is the detailed matching orders:

Order	Matched Price	Matched Volume
1	$p_1 = 60{,}200$	$s_1 = 3{,}000$
2	$p_2 = 60{,}300$	$s_2 = 2{,}000$
3	$p_3 = 60{,}400$	$s_3 = 2{,}000$
4	$p_4 = 60{,}500$	$s_4 = 1{,}000$
Total matched volume		$\sum s_j = 8{,}000$
Total matched value		$\sum s_j p_j = 482{,}500{,}000$

Implementation shortfall is as follows:

Delay cost

$= 8{,}000 \times 60{,}100 - 8{,}000 \times 60{,}000 = 800{,}000$

Transaction cost

$= 482{,}500{,}000 - 8{,}000 \times 60{,}100 = 1{,}700{,}000$

Opportunity cost

$= (10{,}000 - 8{,}000) \times (60{,}800 - 60{,}000) = 1{,}600{,}000$

Fees and taxes

$= 8{,}000 \times 200 = 1{,}600{,}000$

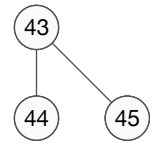
Implementation shortfall

$= 800{,}000 + 1{,}700{,}000 + 1{,}600{,}000 + 1{,}600{,}000 = 5{,}700{,}000$

Implementation shortfall is not always bad. In many cases, they can help us trade at a better price, or trade less and avoid a big loss when there's a delay in trade execution.

However, algorithmic traders should be fully aware of the implementation shortfall and need to measure and evaluate the causes and implement reasonable solutions to ensure the costs don't get out of control. For example, we can reduce the delay costs by placing orders faster after we decide to trade. An algorithmic trading system can optimize the delay to under 60 milliseconds.

In the case of high-volume trading on a market with low liquidity, implementation shortfall would be the top priority. In this situation, we can consider execution algorithms like VWAP and TWAP.



CHAPTER VIII

EVALUATION CRITERIA FOR TRADING ALGORITHMS

43

RETURN RATE

Return rate assessment is the process of measuring the profitability and the risk of a trading algorithm over a specific time period. It can be used to answer the following questions:

- Is the return rate high or low, compared to which benchmark?
- Is the algorithm taking too much risk to increase its chances of making a profit?
- Is the algorithm profitable due to skill or luck?

Relative Return Rate

The relative return rate is the rate of return on a portfolio relative to a benchmark. The benchmark is selected depending on the nature of the algorithm. For example, a trading algorithm that trades underlying stocks may use the VN-Index benchmark, while a market-neutral algorithm may use the risk-free rate instead.

For example, algorithm A trades underlying stocks under a smart-beta strategy, while algorithm B trades on a market-neutral approach. During the same 1-year period (Jan 1, 2021, to Dec 31, 2021), both algorithms achieved a return rate of 20%.

In absolute terms, the two algorithms are equally profitable. However, in relative terms, there's a big difference. It's due to the two algorithms having different benchmarks. For algorithm A, the benchmark is the VN-Index, which actually increased by about 34% during the comparison period. For algorithm B, the benchmark is the 3% risk-free rate.

The two algorithms have equal absolute returns but different relative returns. Algorithm A has a worse result than the benchmark (20% - 34% = -14%), while algorithm B has a better result than the benchmark (20% - 3% = 17%). This difference is most evident in bad years in the market.

Sharpe Ratio – Return on Risk

The main goal of investing is to maximize returns. However, the higher the return, the higher the risk. Therefore, when evaluating the rate of return, it's necessary to consider the risk factor altogether.

The Sharpe ratio, named after the American economist William Sharpe, can solve this problem by dividing the return rate by the risk parameter. This allows for the evaluation of the investment performance on the basis of balancing between return and investment risk.

The formula for the Sharpe ratio is as follows:

$$\text{Sharpe ratio} = (R_p - R_f) \div \sigma_p$$

where:

- R_p is the portfolio's return;
- R_f is the risk-free return rate;
- σ_p is the portfolio's standard deviation, representing risks.

The Sharpe scale rating threshold is as follows:

< 1	1 – 1.99	2 – 2.99	≥ 3
Not good	Average	Good	Excellent

In the algorithm optimization stage, it's necessary to consider whether additional profits are due to rational optimization or

simply taking on higher risks. Instead of focusing on increasing the expected return, investors should focus on increasing the Sharpe ratio. A high expected return is good, but it's only optimal if it doesn't excessively increase the risk.

Skill or Luck

Imagine two investors A and B randomly buy a stock and hold it for a year. Investor A's share price goes up many times while it's the complete opposite for investor B. Does that mean investor A has better investment skills than investor B?

Sometimes, high profits are just a matter of luck. An algorithm that has a stable performance can still suffer big losses due to unforeseen events. On the other hand, an algorithm that has very high profits but has only a few trades may just be lucky rather than effective.

When an algorithm has enough trades, the random factors will cancel each other out. It's important for algorithm traders to distinguish between luck and skill in their trading process. To achieve consistent returns and beat the average market over the long term, skill always matters more than luck.

However, the return rate is not the only aspect to fully evaluate an algorithm's performance. One aspect is the worst-case scenario when running the algorithm. What should be done when the losses exceed the acceptable limit? This issue will be discussed in Article 44 on maximum drawdown (MDD). Another aspect is how to use leverage effectively. When and how much leverage should the algorithm use? The formula to find the optimal leverage parameter will be presented in Article 45 on the Kelly criterion.

44

MAXIMUM DRAWDOWN IN ALGORITHMIC TRADING

When trading stocks, investors are often concerned with the following risk-related questions: "How much money am I likely to lose if the market moves in a direction that is not as expected? How long will it take me to get back to my original position if a loss unfortunately occurs?"

To answer these questions, we can use a concept called drawdown. Drawdown is the percentage of capital lost from a peak to the lowest point before a new peak is reached.

For example, suppose an investor buys FPT stock at 100,000 VND. The stock then rises to 115,000 VND (creating a new high) and then falls continuously to 90,000 VND (creating a new low). The stock then recovers and is currently priced at 117,000 VND (new high). The drawdown in this example is:

$$(115{,}000 - 90{,}000) \div 115{,}000 = 21.74\%$$

Capital	Drawdown	Capital Remained	Required Return Rate to Recover
100	10%	90	11,11 %
100	30%	70	42.86 %
100	50%	50	100 %
100	70%	30	233.33 %
100	90%	10	900 %
100	100%	0	Unable to recover

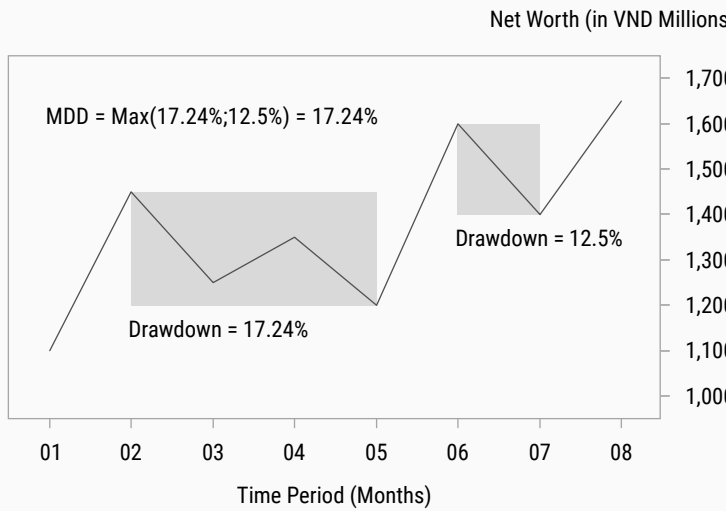
Measuring drawdowns and building a solid capital management strategy is a key factor when building trading algorithms. The larger the drawdown, the higher the risk. At that time, it becomes more difficult and time-consuming to recover lost capital. There may even be no opportunities to return to the original position.

Maximum Drawdown (MMD) In Algorithmic Trading

In algorithmic trading, investors can continuously open and close many different positions. These positions will contribute to an increase or decrease in the total investment. Suppose an investor has 1 billion VND of initial investment to invest in an algorithmic trading system. The invested capital changes over time as in Figure 18.

During the running time of the trading algorithm, there may be different drawdown time periods with different values. The one with the largest value will be called Maximum Drawdown (MDD). In the example above, MDD will be 17.24%. MDD can be understood as the worst loss situation occurring in the timeframe under consideration.

Figure 18 An Example of MDD from an Investment



How to Use MDD

The theoretical MDD from backtesting can be used to estimate the potential risk of the algorithm in real trading. However, if the real MDD exceeds the backtest MDD significantly, it means that the latter is no longer reliable. In that case, investors should pause the algorithm to review its performance.

At ALGOTRADE, we will stop and review an algorithm when its real MDD reaches 150% of the backtest MDD. This is based on the assumption that the historical data used for backtesting may not fully capture the long-term performance.

The table below presents the MDDs of our live algorithms, as of Apr 1, 2023.

Algorithm	MDD
Genesis	-12.48%
Moon	-51.88%
Thor	-29.93%
Polaris	-35.52%

From these MDDs, we can see that the “Moon” algorithm involves the highest risk, while the “Genesis” algorithm is the safest. When allocating capital to different algorithms, investors can use MDD as a parameter to make decisions based on risk tolerance and return expectations.

45

KELLY CRITERION: DEFINITION AND APPLICATIONS

The Kelly criterion is the optimal ratio of assets to risk on each trade to maximize the long-term results. Different algorithms may need different amounts of capital or different amounts of leverage to perform well. The Kelly criterion shows that it's not always optimal to use 100% of our assets on every trade.

Kelly's Formula

The Kelly criterion was first developed for a game to find the optimal ratio to bet. In this game, you either double your money or lose it in every bet. Suppose you have a 60% chance of winning on each bet and you can bet 300 times. Intuitively, if you bet all your capital every time, you would go bankrupt sooner or later even with a positive expected return on each bet. The optimal bet should thus be less than 100%.

The formula for the best bet is:

$$f^* = p - q \div b = p - (1 - p) \div b$$

where:

- f^* is the optimal percentage to bet;
- p is the chance of winning;
- q is the chance of losing ($q = 1 - p$);
- b is the ratio between the amount won to the amount lost.

We can explain this formula as follows:

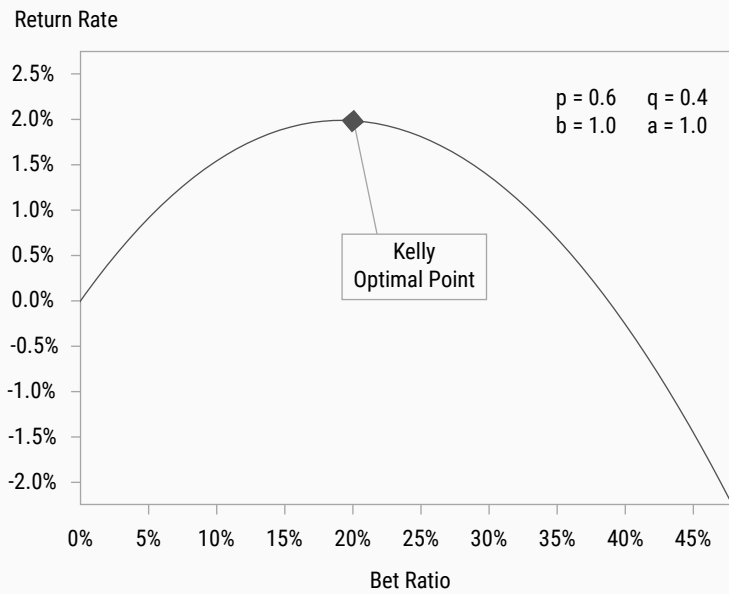
$$\text{Kelly(\%)} = \text{Chance of winning} - \frac{\text{Chance of losing}}{\text{Amount won} \div \text{Amount lost}}$$

Using the formula for the example above:

Kelly(%) = 60% − 40% ÷ 1 = 20%

So betting 20% of your total money is the best strategy in this example.

Figure 19 Example of the Kelly Optimal Point to Bet



Note that this formula implies that you should never bet on anything with a negative expected return.

Investment Formula

Investing often involves partial losses, unlike gambling where the player may lose everything. For this reason, there is a need for a more general form of the Kelly criterion in the investment field:

$$f^* = p \div a - q \div b$$

where:

- f^* is the optimal percentage to bet;
- p is the chance of winning;
- q is the chance of losing (q = 1 − p);
- a is the percentage of capital lost when losing;
- b is the percentage of capital gained when winning.

One of the challenges of using Kelly is that it requires knowing the exact return on investment for each scenario, which is very hard to do in investing. Therefore, the Kelly criterion is not very popular among investors.

Kelly Criterion for Algorithmic Trading

According to the law of large numbers, algorithmic traders can estimate the long-term outcomes of their investments, such as expected return, MDD, winning rate, losing rate, and the profit or loss ratio in each case. These parameters enable the use of the Kelly criterion.

For example, after carefully backtesting and forward testing, an algorithmic trader finds out an algorithm has the following performance and expectations:

Chance of winning	50%	Winning amount	2,00%
Chance of losing	50%	Losing amount	-1.84%

Intuitively, if two consecutive trades have 1 successful and 1 unsuccessful trade, this algorithm will make this profit:

$$(1 + 2\%) \times (1 - 1.84\%) - 1 = 0.12\%$$

This profit margin is small, so the trader uses maximum leverage to maximize their profit. In the Vietnamese derivatives market, they use the default leverage of 5, hoping to make a profit of 0,12% x 5 = 0,6% after one losing trade and one winning trade.

After a year of trading, they keep losing without knowing the reason. The Kelly criterion is the answer.

Kelly Criterion	Expected Profit After 3000 Trades (%)
0,5	+189.1
1,0	+ 534.0
1,5	+ 955.0
2,0	+1232.7
2,5	+1177.9
3,0	+830.3
3,5	+414.0
4,0	+115.5
4,5	-31.5
5,0	-83.5

A good algorithm turns into a losing investment by using the default leverage on the Vietnamese derivatives market.

To find the optimal leverage parameter, Kelly's formula can be applied as follows:

$$f^* = 50\% \div 1.84\% - 50\% \div 2\% = 2.17$$

Using the Kelly criterion as above, the algorithmic trader will make a significant profit instead of a loss, while only using parts of the leverage. This approach can also benefit the entire trading system, as it allows another algorithm to take advantage of the unused capital.

Note that using too much leverage will increase borrowing costs and transaction fees. It will negatively affect the final performance of the algorithm.

The Kelly criterion is a decisive parameter for algorithmic traders in gold, forex, and crypto markets. In these markets, the leverage can be up to 500 times the invested capital.

CHAPTER IX

OPTIMIZATION IN MULTI-ALGORITHM TRADING

46

CAPITAL OPTIMIZATION

Optimization is a common topic to improve investment results. But how can traders optimize capital allocation among different real-time trading algorithms in real-time? This article explains an approach to optimize that in real time.

Basic Approach

A simple way is to allocate a separate account for each algorithm so account statements reflect the complete picture of respective algorithms. This method is straightforward and suitable for traders who use only one single algorithm. The need for multi-algorithm optimization arises when running multiple algorithms at the same time.

To illustrate this point, let's look at an example from the Vietnamese derivatives market:

Algorithm	Current Portfolio	Balance (Number of Contracts)
A	20 long contracts	20
B	30 short contracts	10
C	0	40

Using the basic approach, we would need three separate accounts with a minimum of 40 contracts each, for a total of 120 contracts.

Integrated Approach

Another way is to merge all three algorithms into one single account. This way, after integration it will hold 10 short contracts (20 long contracts and 30 short contracts) instead of 50 contracts. In addition, the minimum capital will be reduced from 120 to 80 contracts. The calculation is as follows:

$$\begin{aligned}\text{Minimum capital} &= \text{Current portfolio} + \text{Capital balance} \\ &= 10 + (20 + 10 + 40) = 80\end{aligned}$$

Comparison Between the Basic and Integrated Approach

The integrated approach can achieve the same performance as the simple approach while lowering the initial investment by 33.33%. This is a significant advantage in algorithmic trading.

	Basic Approach	Integrated Approach
Number of accounts	3	1
Current portfolio	50	10
Minimum capital	120	80

What if all three algorithms at the same time have long positions? How does the integrated approach handle this situation?

Optimization

To optimize capital allocation, we need to answer the questions in the table below:

Operational Capacity	100%	99%	95%
Required capital	?	?	?
Expected return according to the basic approach	?	?	?

In this context, 99% of the operational capacity means we are willing to miss one trade out of every 100 trades.

To get the right answers to the questions above, we need to check all the past trades. The trick here is to trust the law of large numbers and not stress too much about missed trades. A system that has a 20% expected return at 100% capacity will yield a 19.8% expected return at 99% capacity.

For instance, if an algorithmic trader gives these answers:

Operational Capacity	100%	99%	95%
Required capital	120	80	60
Expected return according to the basic approach	20%	19.8%	19%

We can estimate the system's expected return as follows:

Operational Capacity	100%	99%	95%
Expected system return	20%×(120÷120) = 20%	19.8%×(120÷80) = 29.7%	19%×(120÷60) = 38%

This indicates that capital optimization has a major impact on the overall account performance.

ALGOTRADE currently opts to operate at 99% capacity. This decision ensures that the system benefits from the concurrent operation of multiple algorithms while reducing the performance loss due to missed trades. The key message is to accept missing 1 – 5% of trades for the greater good of the entire system.

Implementation

- *Accounting feature*. We need an internal accounting system to separate the trade records and the algorithms' performance since they all run on the same account.
- *Risk management feature*. A multi-algorithm account can face a domino effect that can stop the entire system or cause huge losses. We need proper risk management to ensure safe operation of the whole system. Also, we need to prepare for the scenario of zero capital balance of any single algorithm, in case the system keeps sending requests and accidentally causes a DDoS attack on the broker's server.
- *Queuing feature*. Sort signals by priority queue where some signals are listed in the rare 1% – 5% and never executed. In Vietnam's algorithmic trading environment, we can limit latency to 100 milliseconds.

Optimizing the capital for multi-algorithm trading will be a big challenge in the beginning, but the rewards will be well worth it. Professional algorithmic traders should seriously consider this optimization to scale the system exponentially.

47

BETA OPTIMIZATION

Imagine your system trades a lot of positions on a typical day. Then a rare "black swan" event happens. You realize the system has accumulated too many positions in the same direction, and your total capital is severely affected. How can you prevent such events in the future? This article will help you through these situations.

Prerequisites

The system must be able to generate alpha for beta optimization to be beneficial.

Table 04 Beta in the Vietnamese Stock Market

Investment	Lower Bound	Upper Bound	Note
Stocks	0	2	1:1 loan allowed at 9% - 14% per annum interest.
Future contracts	-5	+5	500% leverage default by law. Higher leverage exists in the market.

These margins can be legally done through all major brokers in Vietnam.

Beta Optimization

A professional trader in Vietnam usually has an absolute beta between 1.5 and 3. We recommend an optimal beta range from 0.8 to 1.2, with an acceptable beta range is 0.5 to 1.5. This metric, though not mathematically proven, can be estimated on the following bases.

First, the market always grows in the long run due to economic growth. At a conceptual level, a long position is more profitable in the long run than a short position. We favor the algorithm with more chances of opening a long position to take advantage of the general market beta. While big losses do happen in times of crisis or catastrophic macro events, it's the risk that a system must accept. The beta strategy achieves about a 7% annual return in the Vietnamese stock market. This return rate is similar to a savings account, while there are more risks in the stock market. However, the assumption of a 7% annual return supports the concept of a positive beta system.

Second, keeping beta between 0.8 to 1.2 makes the system more resilient to unexpected black swan events. The system can withstand any market crash unless all assets become worthless.

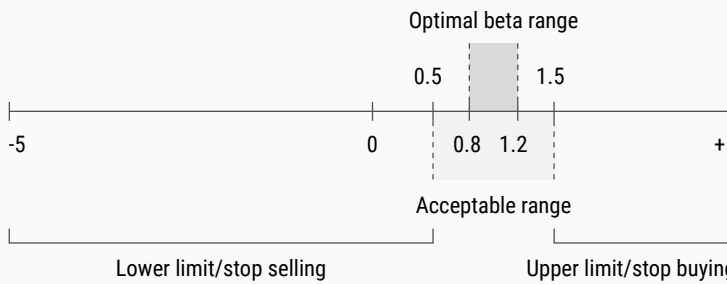
Third, the optimal beta helps reduce margin costs. In the long term, borrowing is not a good strategy. As of 2022, the margin rates in Vietnam range from 9% to 14% per annum.

Finally, with a beta of 1 (and a portfolio split evenly between stocks and derivatives), there's still 40% of the capital left for other strategies such as the market-neutral strategy. The alpha from these strategies from this 40% capital can boost the performance in tough times and diversify the system further.

Suggestions When the Beta Is Outside the Optimal Range

- *Below lower limit*: prioritize algorithms that favor long positions and avoid opening more short positions.
- *Above upper limit*: prioritize algorithms that favor short positions and avoid opening more long positions.

Figure 20 Optimal Beta Range



In simple terms, when the system overbuys, it will ignore any coming buy signals and only accept neutral or the opposite signal. It will miss out on profits from a long position while avoiding catastrophic loss from a black swan event.

When the system has many long positions or short positions, with the same winning probability, opening a position in the opposite direction will be much more profitable in the long run.

48

LEVERAGE TRANSACTION DATA

Importance of Trading Data in Vietnam

In the Vietnamese derivatives market, API data, live boards from data providers are not actual order matching data.

The table below is an example data source that returns VN30F1M at 11:29:58, matching 820 contracts at a price of 1082. This data can be interpreted as around 11:29:58, there are 820 contracts matched at the 1082 range. The standard margin of error is usually around 0.3 points when the market has low fluctuation. However, it can go up to several points in the case of high volatility. For example, the deviation can be up to 30 – 40 points in the case of "call margin".

The cause of this deviation is due to the phenomenon of "tick aggregation" – adding up many ticks in a period of time and returning a single tick. In Vietnam, the ticking time usually falls in about 02 seconds. The data returned can also be delayed by several minutes or even be unavailable for a period of time due to usually unpublished reasons. This means that, even with access to the best data source in Vietnam, the data still has deviations from reality. Therefore, traders may consider using algorithm trading data to collect hidden data with absolute precision.

Table 05 "Tick Aggregation" Example

	Real Matching		Received Data	
Time	**Price**	**Volume**	**Price**	**Volume**
11:29:56:134	1081.5	19		
11:29:56:466	1081.6	3		
11:29:56:519	1082.0	144		
11:29:56:772	1082.5	200		
11:29:56:983	1082.4	38		
11:29:57:431	1082.3	40		
11:29:57:522	1081.9	135		
11:29:57:733	1081.9	30		
11:29:57:738	1081.7	201		
11:29:57:962	1082.0	10		
11:29:58:000			1082.0	820
Total Volume		**820**		**820**

Conditions to Collect Trading Data

The necessary condition is a high-frequency algorithmic trading system. The data is returned continuously enough to reflect the general market, especially at sensitive times when the actual matching price and the hypothetical price highly deviate.

The sufficient condition is that the system should have enough algorithms. Algorithm diversity will picture a market overview, thereby detecting favorable market conditions for each

algorithm and vice versa. Furthermore, trading algorithms need to have a variety of strategies to avoid collecting the same data. Various trading strategies with different order types will enrich trade data. A typical example is the momentum strategy will give market order execution data while market-making strategy will give limit order execution data.

Practical Application

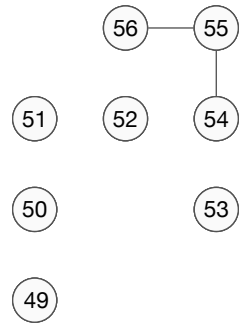
The first application of order matching data is to estimate the extent of market slippage. From thousands of orders matched, slippage can be calculated with high confidence. It helps increase the reliability of backtesting and paper trading. It should be noted high slippage is detrimental for algorithms using market orders but advantageous for algorithms with limit orders.

Traders should also get statistics of "sensitive" times of the market when the algorithms operate at a completely different speed compared to the average. At these times, the transaction frequency can be up to 10 times higher than normal. Any system improvement will bring outstanding results for automated trading.

Another important application of order-matching data is to form market expectations, thereby optimizing capital allocation to different trading strategies. An example is to balance allocation between the momentum trading strategy and the mean-reversion strategy.

The last is to catch unusual transactions in the market, bringing operational advantages to the system. The principle is that if an unusual event occurs frequently, it's likely an important signal to follow.

Combining market and trading data helps ALGOTRADE adapt to different scenarios. For example, our system still operated reliably when live boards and market data in Vietnam were frozen during the Covid-19 pandemic.



CHAPTER X

ALGORITHMIC TRADING PRACTICE

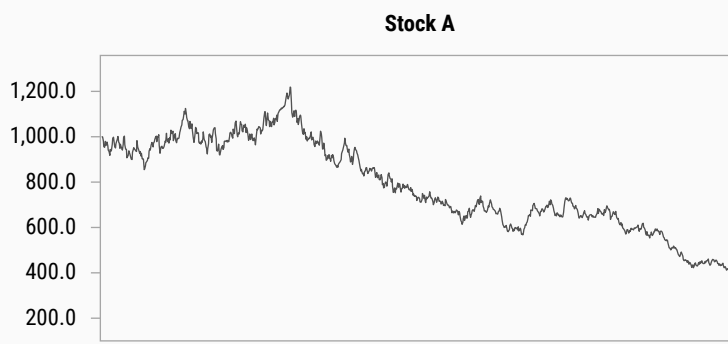
49

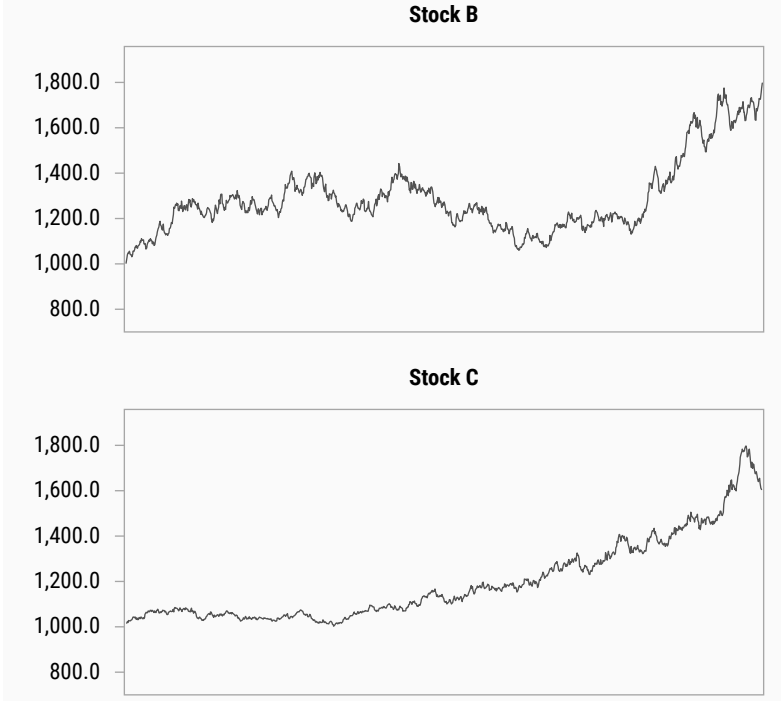
PATTERNS AND RANDOMNESS

In finance, it's difficult to distinguish between skills and luck. In the short term, luck can account for up to 95% of total results. In the long term, however, skills will be 95% of the total results instead. Proven algorithms which eliminate luck or randomness in the long run are important goals for algorithmic traders to succeed.

Here are the 3 price charts of stocks A, B, and C over the last 1000 days. In your opinion, which stock will trend up, trend down, or trend sideways?

Figure 21 Comparison and Identification of Stock Trends





Mark your answer below and see the right answer.

Algorithm	Uptrend	Downtrend	Sideways
Stock A	☐	☐	☐
Stock B	☐	☐	☐
Stock C	☐	☐	☐

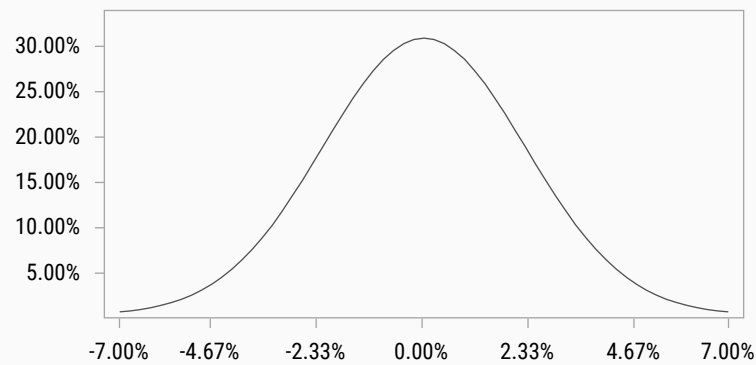
Do you think stock A will go sideways or trend down without recovery?

Do you believe stock B will rise strongly?

Do you believe stock C will move into a correction in the near term?

The final answer for all three stocks cannot be predicted. It's because the three graphs above are purely from a lab simulation. It simulates the next day's price is normally distributed around the previous one, ranging from -7% to 7% (according to the HOSE price limit).

Figure 22 Normal Distribution of Stock Price Daily Movement



The truth is the next day's price is random, unpredictable, and independent of the previous 1000 days in all graphs above. This is a prime example of a striking similarity when comparing price changes by model and by randomness.

The Danger of Confusing Patterns and Randomness

If an algorithmic trader finds a winning model from thousands of stocks like A, B, and C, what would the long-term outcome be?

According to the law of large numbers, no matter what models were discovered, investors will have their assets slowly decline due to the effect of fees, taxes, and slippage. This is the main reason why most passive trading funds are more efficient than active funds, with management and trading fees included. In other words, in a stochastic market, excessive trading can only lead to excessive losses, not including immeasurable fees.

In the financial world, the biggest reason that causes the confusion between a predictive model and randomness is that wrong investments can still make profits and vice versa. In other professions such as musicians, pilots, engineers, and artists, the right decision will usually have corresponding positive feedback and vice versa. In finance, the feedback typically has lots of noise. Following a few specific cases can lead investors in the wrong direction. What if you make bad investment decisions but still get profits five times in a row? Or conversely, what if you make great decisions but suffer losses five times in a row?

This confusion can bring extremely negative results in the long run. As an example, inexperienced investors with no knowledge of the stock market can still make profits as large as multiple times of their investment assets. At that point, they may feel like they already identified the right investment model and start using margin and debt instruments, or provide investment authorization services. However, when a catastrophe happens, these investors may lose all the profits accumulated from the previous period.

This investment phenomenon was widely popular during the Covid period when new investors entered the market and accidentally made huge profits in just a very short time. This pattern will likely continue in the future and is not limited to the stock market.

How to Identify Predictive Models

Distinguishing patterns from randomness is a complicated process since they both appear exactly the same. Even randomness may yield better results in the short term. There are two main methods to identify patterns and avoid randomness in algorithmic trading:

First, use multiple rounds of rigorous testing with a large number of transactions to ensure that the odds to gain long-term profits after all fees and taxes are much greater than the 50% threshold. Algorithmic traders need to understand the steps of backtesting, optimization, and forward testing for detailed implementation.

Second, use economic fundamentals in forming the algorithm hypothesis. A profitable algorithm is not everything. It needs a solid explanation. For example, a stock from a well-performing company with trustworthy reports is much better than a sharply rising stock for unknown reasons. In the long term, algorithms using machine learning may grow beyond human understanding. However, in the short term, understanding the source of profits will ensure the system operates stably and builds sustainable growth.

50

IS ALGORITHMIC TRADING A ZERO-SUM GAME

A zero-sum game is a competitive scenario where one side's gain is equal to the other side's loss. In algorithmic trading, for example, if one investor makes a profit, does that always mean that other investors will lose money?

Is Stock Investing a Zero-Sum Game?

Before we get to the key question, let's take a look at a hypothetical stock investing scenario as follows.

Five investors A, B, C, D, and E pooled their capital to start an ALGOTRADE company, with an initial capital of 100 billion VND. Each invested 20 billion VND and owned 20% of the company's shares.

Three days later, A decided to sell all his shares to G. A agreed to sell at a loss for just 18 billion VND to speed up the process.

A few days later, G found H, who was optimistic about ALGOTRADE's potential, and offered to pay 23 billion VND for all 20%, and G agreed to sell.

Investor	A	B	C	D	E	Total
Investment capital	20	20	20	20	20	100
Shares	20%	20%	20%	20%	20%	100%

In just a short term, nothing special happened in the business and the total value of ALGOTRADE remained at 100 billion VND. G made a short-term profit of 5 billion VND, while A lost 2 billion and H lost another 3 billion due to paying 23 billion VND for an asset worth 20 billion VND only. The sum of all six investors was zero.

Investor	**A**	**B**	**C**	**D**	**E**	**G**	**H**	**Total**
Purchase value	20	20	20	20	20	18	23	
Current price/value	18	20	20	20	20	23	20	
Profit/Loss	**-2**	**0**	**0**	**0**	**0**	**+5**	**-3**	**0**

Two years later, ALGOTRADE prospered and the company's value increased 150% to 150 billion VND. The share price of the investors thus rose accordingly.

The cash flows of all six investors over these two years were as follows:

	A	**B**	**C**	**D**	**E**	**G**	**H**	**Total**
At the start	-20	-20	-20	-20	-20			-100
A few days later	+18					-18		
						+23	-23	
Two years later		30	30	30	30		30	150
Total Cash Flow	**-2**	**+10**	**+10**	**+10**	**+10**	**+5**	**+7**	**50**

After two years, the total net benefit of all six investors was 50 billion VND, even though A still lost 2 billion VND.

If there are no major events in the short term like war, natural disasters, etc., the core value of a business hardly changes in a few days. Buying and selling stocks for short-term gains can be seen as a zero-sum game.

In the long term, if there are macro and micro events directly affecting the business fundamentals, the total welfare of the investors will be either greater or lower than zero depending on the events. In other words, investing in stocks over the long term is a non-zero-sum game.

From an optimistic perspective, the profits of publicly listed companies tend to grow over time along with general macroeconomic growth. Investors will benefit from the strong performance of the business – a game with a sum greater than zero where everyone wins.

Is Derivatives Trading a Zero-Sum Game?

Ignoring taxes and fees, VN30F futures trading is a zero-sum game because all of the trading cash flow is not put into businesses, only transferred from losers to winners. In fact, if all fees and taxes are included, it results in a negative-sum game.

Is Algorithmic Trading a Zero-Sum Game?

Whether algorithmic trading is a zero-sum game depends on the strategy used. Some strategies are based on short-term movements like scalping and day trading. These are prime examples of a zero-sum game. In the long term, without a clear alpha, these strategies will lose money because of taxes, fees, and price slippage.

Most other investment strategies have a positive sum, where the total benefit is greater than zero. A good example is long-term investing using algorithms that expect to profit from the

overall growth of the economy. This is a positive-sum game, where everyone wins in the long run.

Moreover, when the market is inefficient, information is not updated fully, quickly, and transparently, and the process of evaluating and analyzing information becomes inaccurate. The process of making investment decisions becomes inefficient. As a result, the stock price deviates significantly from its true value. It results in huge wins and heavy losses. It's also a zero-sum game but with a wide gap between winners and losers. This can have negative consequences for the overall stock market and the economy: liquidity decreases, good businesses have difficulties raising capital for production and cannot create added value in the long term. Algorithmic trading strategies like arbitrage, market-neutral strategy, etc. can help increase the market liquidity and reduce the gap between winners and losers in the short term, making the market more efficient. When the economy grows steadily and sustainably, all investors will win in the long run.

The execution algorithms that reduce trading costs in large-volume transactions are also not part of the zero-sum game.

Especially, the market-making algorithms help increase liquidity, match trading orders, and also contribute greatly to the development of the general market.

In conclusion, an investor's approach will decide whether they are participating in a zero-sum game or not. Certainly, positive-sum games are always beneficial for investors in the long run.

51

WHAT TO DO WHEN IN DOUBT OF TRADING ALGORITHMS

The market appears to plummet, but your algorithm is about to likely open long positions. Would you decide to interfere with the system and place orders manually?

You may be tempted to override an algorithm and trade manually when the market seems to go against it. This is a common mistake for inexperienced algorithmic traders. You implemented the algorithm, and you may think you know better than the algorithm does. In reality, however, interfering with an operating algorithm may do more harm than good.

Four Consequences From Interfering With the Algorithm

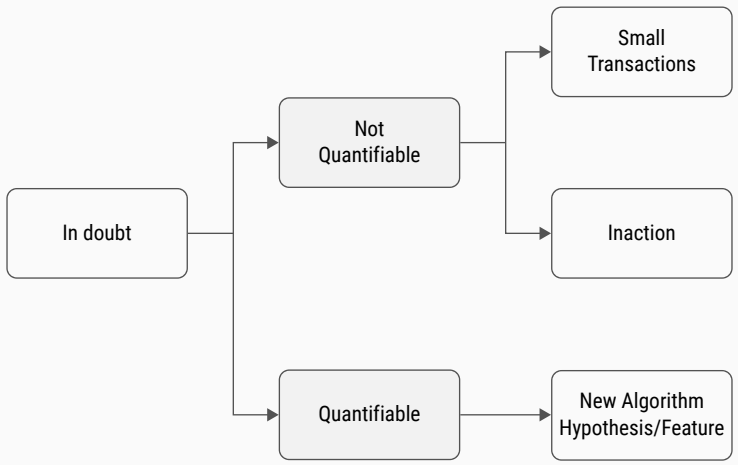
- *Drive emotional trading*. Algorithmic traders leverage computers and tested algorithms to make rational decisions. If you abandon this advantage for a gut feeling, you are going against the essence of algorithmic trading. Every time you interfere with the automated system, you move further away from becoming a true algorithmic trader. You should always be careful not to cross the line between intervention and non-intervention. In the long run, fully trusting the algorithm can help you build a successful career. It will also motivate you to be confident and thorough in every algorithmic aspect. This mindset will benefit you greatly in the future.

- *Increase your stress and reduce your self-confidence*. There are two possible scenarios when you stop the algorithm and open a position in the opposite direction. If you make a profit, you will lose confidence in the algorithm. This is frustrating since you will have to decide every time your feeling differs from what the algorithm does. Should you let the algorithm run or manually intervene as you used to? If you make a loss, you will waste your efforts and feel regretful. The short-term results may vary, but the long-term results are the same: you become stressed and lack confidence.
- *Reduce your free time*. The biggest reward that algorithmic traders get is free time. If you give up automated trading for manual trading, you will have to spend all your time monitoring the algorithm to intervene at the right moment. Whether you make a profit or not, you are giving up the best benefit of algorithmic trading.
- *Invalidate your statistics*. What do the statistics of your algorithm mean after interventions? Nobody knows because the data is no longer reliable. This will make it very hard to evaluate and improve your algorithm.

What to Do When You Doubt Your Algorithm

The answer is to try to exploit new ideas without interfering with the running algorithm. Below's a model to explain this concept.

Figure 23 What to Do When You Doubt Your Operating Algorithm



If your doubt is purely emotional and not quantifiable, you should consider doing nothing. Sometimes, the doubt may persist. One trick to help relieve emotional pressure is to make small trades on another account. This way, you can satisfy your emotion and keep the integrity of your algorithmic data, while also having manual data for analysis.

If your doubt is quantifiable, you should save the idea and consider it as a potentially new algorithm, or a new feature for future development.

This approach preserves valuable information from your doubt if it exists while keeping the running algorithm untouched. In the long run, this will be very helpful for harnessing both your creativity and your trust in your algorithms.

52

SCAMS IN ALGORITHMIC TRADING: 07 MAJOR CHARACTERISTICS

Some scammers often use high-tech terms like neural networks, artificial intelligence, machine learning, and market sentiment in algorithmic trading to deceive naive investors and make unrightful income. This is quite common in Vietnam's stock market. Investors can save a lot of money by knowing the 07 key features of scammers in algorithmic trading.

1. 100% Success Rate

The first rule of a scammer is to appear perfect to the victim, never losing in any trade. This creates an illusion for investors that winning is easy. If an investor sees someone claiming or showing themselves as an absolute winner in any way, they have probably found a scammer.

2. Fail to Provide a Complete Statement

A scammer cannot have a good transaction statement, so they cannot provide this information. At best, the scammer can only show some profitable trades but never a complete statement.

3. Always Announce Position Opening/Closing Points Late

The scammer cannot provide real-time position opening/closing points because they do not exist in the system. Instead, they announce the positions after the fact to ensure a 100% success rate.

Figure 24 An Example of Scam Announcement After the Fact

A typical example is that in a bull market scenario . A scammer will post a chart image like the one above with colorful indicators and claim: bought VN30F1M at 1045.0 points and imply an unrealized profit of 10 points.

The result of this late announcement in position opening/closing is that the unrealized attractive profits are never executable.

There's a term for this type of scammer: "time machine" traders. They are traders who can buy and sell in the past.

The stereotype for this group is as follows:

- If VN30F1M increases by 5 points, they announce they purchased at the current VN30F1M price minus 5;
- If VN30F1M decreases by 5 points, they announce they shorted at the current VN30F1M price plus 5.

4. Hyperactivity

Scammers need to constantly impress and reassure their investors with their fake ability to generate high returns and consistent wins. To keep up the illusion of stability, scammers will devote most of their time to providing false information that shows a steady win every day. In some extreme cases, scammers will claim to have made a profit every 15 minutes or less.

5. Lead to Internal Channels

Scammers rely on "service fees" as their income source. They have to persuade inventors to join some exclusive private channel. In Vietnam, these channels are usually chat groups like Zalo or Telegram. Scammers will use various tactics within the private chat group to extract "service fees" from investors. Some other scammers may offer complete investment authorization services where investors may never get their investment back.

Note that in private channels, about 95% of all accounts are fake and are controlled by scammers to create a bandwagon effect. This effect is meant to convince the victim of the scammer's superior skill and make it easy for the victim to fall for the scam.

6. Personal Attack Right After Being Criticized

A perfect image is crucial for attracting more victims, so any criticism will be taken very personally by the scammer. This is a hidden feature that investors can test to see how scammers react. Remember scammers often control multiple accounts to ensure their majority in any argument.

7. No Programming Skills

Since scammers spend most of their time and energy in luring naive investors, scammers won't have any time to learn or

practice any programming skills. This group usually has no programming knowledge or skills at all. A simple direct question on programming will usually be ignored.

The philosophy of scammers is to make naive investors believe in the possibility of making money quickly without any risk. By understanding the philosophies of algorithmic trading scams and the characteristics of scammers, investors new to algorithmic trading can avoid falling into these traps.

53

THIRD-PARTY SOFTWARE IN ALGORITHMIC TRADING

In Vietnam, algorithmic traders can be divided into 2 groups. The first trades via third-party software such as AmiBroker, MetaTrader, and TradingView, in the order of popularity. The remaining built their own system, from as simple as Excel, to as complex as Python, C, and Java. The third-party software group accounts for about 80% of the algorithmic trading community in Vietnam.

About AmiBroker

AmiBroker is a user-friendly software that only requires basic programming skills. It can be used by different types of investors, from beginners to experts in algorithmic trading.

The software can handle various types of data sources. The easiest one is historical data that can be used to backtest trading strategies. Another option is to buy real-time data from vendors for an annual fee of around 01 to 05 million VND. However, this option may have some drawbacks such as high latency and data errors. Alternatively, traders can collect their own data for free and make the program themselves with low latency using the AmiBroker DLL plug-in.

AmiBroker is designed specifically for backtesting and technical analysis. The software provides many tools to analyze

market data and calculate technical indicators. The advantage is the speed of testing and optimization. To fully automate the trading system, users need to use APIs provided by securities companies. A disadvantage of AmiBroker is that the standard format for input data only includes OHLCV, Aux1, Aux2. Other factors such as financial statements, market sentiment, news and events are not taken into account in technical analysis. However, users can directly fill in the parameters on the AFL files (AmiBroker Formula Language – a programming language designed by AmiBroker) or import the TXT file to add data.

Let's compare the pros and cons of using third-party software versus building a Python and API system for algorithmic trading.

Pros

- *Reliable*. These software are used by hundreds of thousands of traders around the world, so they are very stable and dependable. Investors can trust them to run smoothly and consistently.
- *Time-saving*. Building a trading system that covers all the basic functions can take years or even never be completed. With AmiBroker, MetaTrader, and TradingView, investors can have a ready-made system in a few hours at a reasonable cost.
- *Excellent for technical analysis*. These software originated from foreign exchange market (Forex), so they have great features for technical analysis traders who use price and volume data as inputs.

Cons

- *Hard to customize*. These software are designed for the global market. Any changes to adapt to Vietnamese market conditions will be challenging.

- *Lack of fundamental data for the Vietnamese market*. Information on financial reports of companies listed in Vietnam is usually not fully integrated. If investors want to carry out in-depth fundamental analysis, AmiBroker and MetaTrader will unlikely be satisfactory.
- *Hard to access account positions*. Investors can only access account information through the API, so AmiBroker and MetaTrader cannot automatically know the account status, which may lead to many unnecessary trading errors. However, there's the AmiPy plug-in for algorithmic trading, though it requires Python programming skills.
- *Limited support for multi-algorithm trading*. AmiBroker and MetaTrader can only support one algorithm per account.

The table below summarizes the experience of using third-party software versus the Python API system in Vietnam.

Investors can use both AmiBroker and Python to optimize the trading system. Python has many libraries that support data analysis and statistical computation, which investors can use to build predictive models, combine different data sources, and analyze data at high speed. Machine learning is a new trend in algorithmic trading that investors can only accomplish with Python.

The most important factor in choosing an algorithmic trading approach is suitability. Third-party software is geared towards beginners who need a quick and easy algorithmic trading experience. To become a professional algorithmic trader who can develop various trading strategies, investors will need a trading system that can be customized to a high degree.

Table 06 Using Third-Party Software Versus Building a Python and API System for Algorithmic Trading in Vietnam

Criterion	Third-Party	Python/API
Basic algorithmic trading	Good	Bad
Intermediate algorithmic trading	Average	Average
Advanced algorithmic trading	Bad	Good
Backtesting	Simple	Complex
Forward testing	Supported	Supported
Multi-algorithm trading	Bad	Good
Trading execution	Good	Good
Account management	Average	Good
Customization	Bad	Excellent
Fundamental analysis	Bad	Excellent
Technical analysis	Excellent	Good
Data	Premium/Free	Free
Software	Premium	Free
Programming	Minimal	Constant investment
Signal speed	Fast	Fast
System speed	Slow	Fast

54

IS ALGORITHMIC TRADING PREFERABLE FOR ALL TRADERS

In the future, algorithmic trading will dominate the Vietnamese market. However, it will be accessible only to a few investors or institutions. It's still not a tool for the mass market. The following are some types of investors or institutions that can benefit from algorithm trading.

Investment Funds or Proprietary Trading Firms With Large and Frequent Trading Volumes

Investment funds and proprietary trading firms need to execute their investment decisions efficiently and effectively. They face many challenges such as staff costs, management, and most importantly price slippage. An algorithmic trading system will yield similar returns with minimal operating costs.

For example, Dragon Capital needs to sell 50 million HPG stocks and buy 20 million FPT in the Vietnamese market. This task may require several weeks for a trading team to complete. With an algorithmic trading system, a fund manager only needs to input the desired outcome and the system will handle the rest, while providing real-time updates. Moreover, an automated trading system can work on more than 100 stocks simultaneously, which is impossible for human traders.

Index ETFs

Index ETFs follow mathematical formulas to adjust their portfolios constantly to ensure their performance follows the tracking index. This task can be done daily by an algorithmic trading system in the whole market. It can calculate the buy/sell volumes and prices to place orders and generate reports.

Professional Investors With Established Investment Methods

Professional investors have developed their own investment methods. These methods can bring profits to their clients or themselves. They can automate their decision-making and trading processes by using an algorithmic system. The main difference between professional investors and new investors is that they have a stable investment formula that can be programmed, thus leveraging the power of algorithmic trading.

At ALGOTRADE, we use algorithmic trading to rebalance our portfolio by scanning the financial statements of more than 1,600 listed companies. The process usually takes several hours.

Mathematical or Risk Management Experts

Math and risk management experts may already have profitable investment strategies. However, they cannot execute them manually. Some examples that involve math are scalping or pair trading. These algorithms require continuous order placement and cancellation, which can only be done via an algorithmic trading system.

At ALGOTRADE, we use our algorithmic trading system to trade thousands of futures contracts every day with optimal capital allocation.

Algorithmic Trading Systems Are Not for Everyone

New investors without a clear and sustainable investment method may be disappointed by the actual results of an algorithmic system when trying to automate their investment through technical analysis models or indicators. They may be looking for the holy grail to get rich every day but they likely overlook the flaws in backtesting or the costs involved in real trading. These investors should start small and gradually improve their skills. Poor trading performance often comes from unreliable backtesting results or inaccurate cost estimation.

55

HOW TO BECOME AN ALGORITHMIC TRADER

If you want to grow your career as a professional algorithmic trader, let's explore the essentials below.

1. In-Depth Financial Knowledge

You'll need in-depth knowledge of the financial markets as well as an understanding of legal structures, taxes, and applicable fees. It will be challenging to just rely on volume and price data without knowing anything about the market. Algorithmic trading requires lots of effort to customize the system according to your needs. Algorithmic trading is more suitable for those who already can make consistent profits with traditional methods.

Estimated percentage of traders with in-depth financial knowledge: 5%.

2. Software Development Skills

You need to have data skills, such as data collection, data cleaning, and data management. Furthermore, system optimization and server management with high programming skills are critical to building an algorithmic trading system. The more developed the market is, the easier it is to accomplish this.

Note that the system development skills require building an entire system from scratch without third-party software like AmiBroker, MetaTrader, or TradingView.

Estimated percentage of traders with technical backgrounds among the remaining traders: 3%.

3. Motivation

Being a software engineer with in-depth financial knowledge is necessary but may be insufficient. You'll need personal motivation to start and realize the project. There will be distractions like family, health, time, or other life interests.

Estimated percentage of traders with motivation among the remaining traders: 25%.

4. Investment Capital

Investment capital is required to start an algorithmic trading career. Without initial capital, algorithmic traders may not earn enough to cover living costs. In Vietnam, the estimated minimum capital is 3.75 billion VND (around US$160,000) while it would be US$700,000 in the US.

Assume the expected return in Vietnam from algorithmic trading is 15% per annum. The basic living cost is 25 million VND a month, or 300 million VND a year, with an estimated inflation of 7%. The minimum starting capital is:

$$0.3 \div (15\% - 7\%) = 3.75 \text{ billion VND}$$

A similar calculation applies to the US with the following estimates: inflation at 2%, expected return at 12%, and basic living cost at US$70,000. The minimal starting capital is then:

$$70.000 \div (12\% - 2\%) = \text{US\$ } 700.000$$

These numbers are not a significant investment, but for an individual, they are not small either.

Estimated percentage of traders with sufficient investment capital among the remaining traders: 10%.

5. Long-Term Profit

An algorithmic trader needs to make profits in the long run, not just in the short term. Many algorithms do lose their edge over

time and need to be adjusted or replaced. Thus, algorithmic traders need to keep researching and developing new algorithms.

Estimated percentage of traders with sufficient long-term profits among the remaining traders: 35%.

6. No Stability Required

Investors need to be comfortable with uncertainty and randomness. It's challenging to predict the profits in just a month. Without the right mindset, investors may not have enough patience during difficult times. Not everyone with the right tools is ready to become an algorithmic trader.

Estimated percentage of traders with the right mindset among the remaining traders: 50%.

7. Choosing Algorithmic Trading as a Career

This criterion is quite tricky. Investors may have all skills and qualities to be algorithmic traders, but they may have many other opportunities from large corporations. Will they pick algorithmic trading if they have other career options? Some will, and some won't.

Estimated percentage of traders who would stick to algorithmic trading: 20%.

Estimated Number of Algorithmic Traders in Vietnam

Let's make a rough estimate of how many algorithmic traders there are in Vietnam. Assume there are 2.5 million individual traders, which is roughly half of the 4.93 million brokerage accounts in Vietnam as of August 2022:

$$2.500.000 \times 5\% \times 3\% \times 25\% \times 10\% \times 35\% \times 50\% \times 20\% \approx 3$$

Our estimate is that there are only 03 professional individual algorithmic traders in Vietnam or one algorithmic trader for every one million individual traders in Vietnam.

Worldwide, with a population of 7.75 billion, let's assume 5% trades in the financial markets, which results in 387.5 million

individuals. There would be around 387 algorithm traders globally.

Therefore, it can be seen that if one wants to pursue a career in algorithmic trading, it's important to combine the professional skills of many excellent individuals.

56

HOW TO LEARN PROGRAMMING SKILLS FOR ALGORITHMIC TRADER

Many wonder how to learn coding or a programming language. This article will provide directions and give resources to help you learn programming.

Which Programming Language to Learn

The key question is not what programming language you should learn but how to learn to code. The most important factor in learning is the programming mindset. It takes time to develop the right mindset; however, some languages will help you train better than others.

In our experience, Python is the easiest language for beginners to learn and practice because it's most similar to natural language. Some other programming concepts and techniques may be better expressed in other languages like C or C++. Many university programming courses still start with C and C++ for introductory classes. A famous example is the CS50 course from Harvard University.

If you have a class of choice, it's best to follow the programming language taught in the class.

Where to Learn Programming

There are many ways to learn programming such as in computer science majors during college, or vocational schools. Or you

can also take short courses at IT centers. There are also online courses and other materials for self-study. These options will be presented more in-depth below.

Study at formal colleges or after-school centers. These are face-to-face, offline learning. If possible, formal education at a university or vocational school is a great option. However, it's also the most expensive in terms of time and money. This option may not be suitable for those who have worked or studied a different discipline. Short-term courses (from 3 to 6 months), mostly outside working hours, are another option for working professionals. Offline classes help learners interact directly with instructors and shorten the learning time compared to self-study. These formal courses also make it easy for learners to engage in a community of fellow learners for lifelong connections. In summary, offline classes should be taken seriously if possible.

Online courses and online materials. Beginners can head to online platforms that help learn to code directly in the web browser like Codecademy and CodeSchool; online learning communities and online course marketplaces such as Skillshare and Udemy. There are also formal online courses such as Coursera, edX, and Udacity. Each has different pros and cons for learning how to program.

Online platform to program directly in the browser. There are many platforms like this but two stand out: Codecademy and CodeSchool. Both offer a wide range of courses for all levels from beginner to advanced in different programming languages such as Python and Java.

The advantage of these platforms is that learners can try and practice programming right in the web browser, without having to install and configure any programming language and supporting tools on a computer. This makes it easier for beginners to start

programming. However, this option does not help learners understand how a programming language actually works in a computer system. When they need to develop a program that can run on a real system, they will need to learn the software development process and the computer system.

Online course marketplace and online learning communities. Udemy and Skillshare are two great examples of this type. Learners can find many courses related to programming, such as Python, Java, etc. These courses are created by independent content creators. The variety will help learners have more options to find suitable approaches and content for themselves. However, most of these classes are not free, and learners may have difficulty choosing the right courses due to the variety.

Formal online courses. These are the first major open online course platforms (MOOCs –Massive Open Online Courses), launched in 2011 and 2012 by professors from Stanford University (Coursera, Udacity), Harvard University, and MIT (edX).

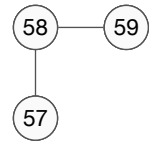
These platforms offer courses from mainstream programs of top universities around the world. These courses can be an alternative to traditional face-to-face classes at university. These platforms have gradually structured their programs to be more relevant to online students.

For example, a course that initially lasts a few months (typically 3 to 4 months) is now broken down into several 1-month courses. So learners can keep up with the pace more easily. This is the closest option to traditional offline university courses. Learners can expect to learn programming from scratch in different programming languages. However, for those who don't like the traditional way of learning, it can be difficult and sometimes boring to follow these courses.

In general, whether it's an online or face-to-face class, learners need to carefully consider their needs and efforts (in terms of time and money) to choose a suitable option. A clear goal, like being able to write a simple Python program in 6 months while working full-time, would be helpful to motivate learners on the path ahead.

What if You Don't Have Enough Time and Money to Learn Programming but Still Want to Pursue Algorithmic Trading?

This is also the key question ALGOTRADE founders have when starting the company. Our short-term goal is to make tools so that traders don't need to know how to program to still start their journey with algorithmic trading. We are working hard to bring this tool to customers in the near future. We'll keep you updated!



CHAPTER XI

INTEL CENTER – SUPPORT CHANNEL

57

INTEL CENTER OVERVIEW

Data plays an important role for algorithmic trading. It helps develop ideas and test algorithm hypotheses. At ALGOTRADE, we build Intel Center that collects and analyzes data that can improve algorithmic trading performance.

Our first version targets the derivatives market, which attracts over 95% of algorithmic traders. The derivatives market is based on the VN30 index. Our Intel Center provides relevant information on the VN30 stocks, the VN30F futures contracts, and other factors that may benefit the trader's investment. ALGOTRADE focuses on providing exclusive information that's not publicly available to give investors a competitive edge.

Intel Center has two main sections, raw data and processed advanced data.

Raw Data

- *Live board*. The live board includes VN30-Index stocks and VN30F futures contracts with basic information such as current price, volume, bid/ask prices, and bid/ask volumes.
- *Real-time prices of major global market indices.* They include the US 500 Cash, Nikkei 225, Shanghai, Hang Seng, KOSPI, FTSE 100 Futures, DAX Futures, and CAC 40 Futures.

Processed Advanced Data

Our live board tracks the trading activities of foreign investors on VN30 stocks to identify market trends. It includes cumulative on-

day trading value together with historical data. Foreign investors' activities on VN30 are expected to be more informative for the derivatives market than foreign investors' activities on VN-Index.

Our Intel Center tracks foreign investors' activities through their real-time holdings of futures contracts. The number of contracts is categorized into 4 types of contracts available in the Vietnamese market. Based on this information and historical comparison, it's possible to estimate the level of optimism and pessimism of investors in the Vietnamese stock market in the medium term.

In future versions, Intel Center will add more features to support traders, such as:

- *VN30-VIX*: an index to measure volatility from VN30.
- *Foreign Activity Tracker*: a detailed tracking tool for foreign investors' ownership in stocks.

In the long term, ALGOTRADE can provide data API to algorithmic traders to access and use this specialized data for their decision making and many other features.

58

FOREIGN TRADING DATA

Foreign Investors

Foreign group registers an account and trades on the Vietnamese stock market. Foreign investors can be individuals, organizations, or foreign investment funds, such as Vinacapital VietNam (VOF), VietNam Holding Limited, and VietNam Equity Fund.

The stock exchange in Vietnam publishes and shares information about foreign investors to all other investors.

Here are key reasons why investors should pay attention to foreign trading data on stocks, especially in the VN30-Index.

- *Stable cash flow*. Foreign investors include ETFs or investment funds from China, Taiwan, Korea, Europe, and America. They usually have large and slow-moving cash flows, which tend to last for a brief period of time. Knowing the net buying or selling trends of foreign investors can help traders limit their risk and increase the expected profit.
- *Large cash flow*. The amount of money that ETFs and funds invest in Vietnam is still small. However, it's still significant compared to the size of the Vietnamese market.
- *Professional cash flow*. The Vietnamese market has many technical barriers and is not fully open to the majority of foreign

investors. Only a very few professional investors or investment funds can access it. Therefore, their trading behaviors reflect the expectations from a special group of professionals.

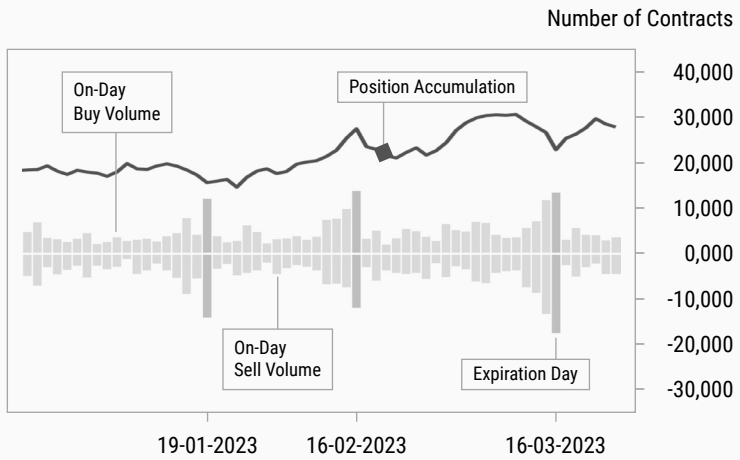
- *Long-term cash flow*. Foreign investors usually buy and hold stocks rather than trading them in a short term within a few days. Foreign transactions typically move the market's valuation into new equilibrium. Even though they have a small share in terms of trading volume, if we consider the long-term investment and ignore short-term transactions, the trading value of foreign investors has a significant impact on the Vietnamese stock market.
- *VN30 is the underlying asset of the futures contract*. The basis (the gap between VN30F and VN30) varies over time, but it needs to converge to 0 on expiration days. Foreign trading data on large-cap stocks of VN30 will thus directly influence the derivatives market.
- *Simulated cash flow*. Some local investors collect foreign trading data to make investment decisions. They mostly hold similar positions to foreign investors. That means when they see a sudden large net trading volume from foreign investors, they will likely follow suit and amplify the impact of foreign investors.

Intel Center provides 2 specialized charts (Figure 25 and Figure 26) related to foreign trading data, focusing on stocks in the VN30-Index and VN30F futures contracts.

Position Accumulation of VN30F From Foreign Investors

Figure 25 below shows the cumulative chart of foreign investors in VN30F and the daily number of contracts they buy and sell.

Figure 25 Position Accumulation of VN30F From Foreign Investors



For example, as of Sep 9, 2022, foreign investors have a total of 14.512 long positions, including 8,465 VN30F1M contracts, 5,494 VN30F2M contracts, 447 VN30F1Q contracts, and 106 VN30F2Q contracts.

This index can help predict the future investment directions of foreign investors because they can buy derivatives contracts first and buy stocks later to limit price slippage. In the example above, foreign investors hold 14,512 long positions of VN30F at 950 points, equivalent to a future contract at 95 million VND. Assume that 70% of these contracts are to hedge the opening price of the underlying securities position. We can estimate foreign investors will be net buyers in the Vietnamese stock market:

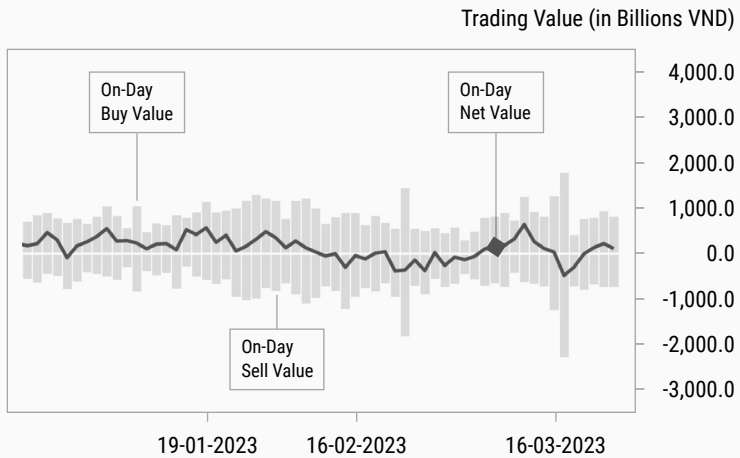
70% x 14.512 x 95.000.000 = 965 billion VND

This estimate gives investors more information to invest in the market.

Trading Value of VN30 Stocks From Foreign Investors

Statistics on foreign transactions is based on the net transaction value in the whole Vietnamese market. However, for traders in the derivatives market, the trading value specifically on VN30 stocks is more meaningful than on VN-Index.

Figure 26 The Trading Value of VN30 Stocks From Foreign Investors



ALGOTRADE created the chart "trading value of VN30 stocks from foreign investors" (Figure 26) to provide real-time foreign transaction data on VN30.

This chart can provide future predictive value since foreign investors are usually stable in the long run. For example, when foreign investors switch from net selling to strong net buying as in the period from November 1, 2022, to November 18, 2022, local investors can expect the trend to continue for many months if market prices stay stable.

Top Stock Changes in Foreign Ownership

Rank 5 stocks that are bought and sold the most by foreign investors compared to the foreign ownership room for each. The formula for calculating the change rate is:

$$\frac{\text{(Buy amount – Sell amount)}}{\text{(Foreign ownership room x Total number of shares)}}$$

The result is a relative value that shows how foreign investors decide for each stock or company. Traders can use these data to determine the net trading trend of foreign investors.

1-Week Period		1-Month Period	
Stock	**Percentage Change**	**Stock**	**Percentage Change**
DXG	2.33%	HSG	13.74%
CMX	1.11%	TNG	4.57%
MIG	1.00%	CNG	4.38%
PAN	0.96%	NKG	3.88%
HSG	0.94%	PAN	3.47%

Top stocks with foreign ownership change (net buy), updated Mar 31, 2023. Visit live.algotrade.vn for latest information.

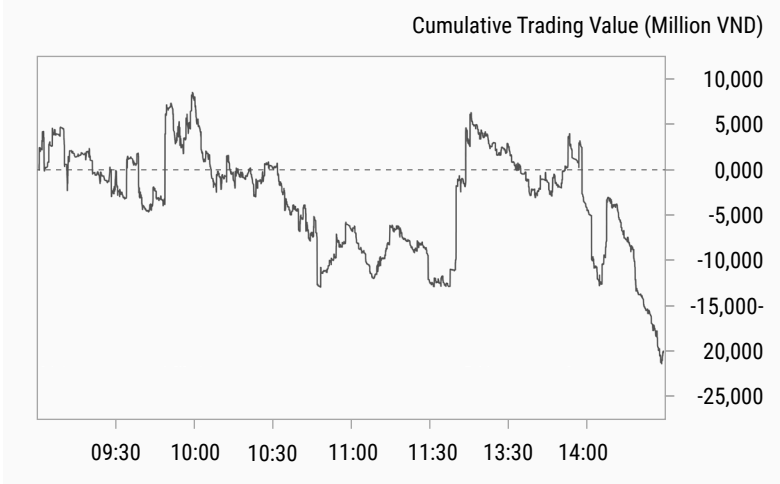
Top stocks with foreign ownership change will be most valuable when looking at a week (5 trading days) or a month (20 days). When looking at a longer term, or much shorter term (like within a day), it's only a reference and does not offer as much information value.

59

DAILY ACCUMULATIVE FOREIGN TRADING VALUE IN VN30

Daily trading value of foreign investors on VN30 can be a key factor in determining investment direction of the derivatives market and the underlying stock market, both in short and medium terms. It's especially important for intraday trading strategies. Below is a reference guide to make use of this value in forecasting market trends.

Figure 27 Intraday Trading Value From Foreign Investors on VN30



Important Thresholds

The thresholds in the table below are collected at the end of a trading day. During trading days, it's possible to estimate the end-of-day value based on simple predictive models.

Absolute Threshold	Interpretation
0 – 50 billion VND	No clear trend
50 – 100 billion VND	Very weak trend
100 – 200 billion VND	Weak trend
200 – 500 billion VND	Good short-term trend
500 – 1,000 billion VND	Strong trend for coming weeks
> 1,000 billion VND	Very strong trend for coming months

In the long run, investors can refer to the following percentage thresholds:

Relative Threshold	Interpretation
0% – 0.2%	No clear trend
0.2% – 0.4%	Very weak trend
0.4% – 0.8%	Weak trend
0.8% – 2%	Good short-term trend
2% – 4%	Strong trend for coming weeks
> 4%	Very strong trend for coming months

Exclude Put-Through Orders

Put-through transactions do not occur through order matching. Their public prices often have no real value, so they don't directly affect market expectations. Currently, in Vietnam, put-through orders are used to transfer assets among legal entities for various purposes outside of trading. The data on put-through orders is noisy and should be excluded to ensure data accuracy.

Combine Cumulative Charts of Foreign Investors' Futures Contracts

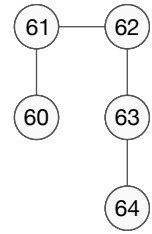
The accumulation of futures contracts from foreign investors is a strong trend indicator because they tend to trade futures contracts before executing stock orders to stabilize the trading price. Thus it's possible to combine the net trading value and the cumulative charts of foreign investors' futures contracts for trading. The table below is a simple example.

Net Trading Value	Accumulation of Futures Contracts	Trend
Strong net buy (> 0.8%)	Increasing	Strong uptrend
	Neutral	Uptrend
	Decreasing	Neutral

In a more complex forecast, it's possible to convert the net trading value into the number of futures contracts to estimate the future cash flow of foreign investors.

For example, foreign investors have a net buy volume of 200 billion VND in the market. Assume VN30 is at 1000 points, which

means each future contract is priced at 100 million VND. The net buy volume is equivalent to 2000 futures contracts. On the derivatives market, if foreign investors sell 500 futures contracts, then their total trading value is still positive at 1500 futures contracts. It can still be seen as a positive sign from foreign investors to the stock market.



CHAPTER XII

ALGOTRADE LAB – FIRST STEP TO ALGORITHMIC TRADING

60

ALGOTRADE LAB OVERVIEW

Building an automated algorithmic trading system for the Vietnamese market in Python or C is challenging. Even with a strong background, it can take up to six months for an average algorithmic trader to build them through APIs.

To bring algorithmic trading closer to investors, we develop ALGOTRADE LAB, a free service that lets you experience algorithmic trading right away even when you don't know how to program. You can connect your personal securities accounts to our system and set up your own trading preferences. The system will handle the rest automatically, and you can intervene or stop it at any time.

With ALGOTRADE LAB, investors can acquire the following experience:

- *Automatic trading*. After successfully setting up API connections, the system will execute your orders based on your setting automatically, unless you opt to intervene or stop the system.
- *Continuous data processing by ticks*. The system will adjust your orders according to every price change in the market.
- *Super fast order execution*. The system will place your orders much faster than traditional trading methods. In some cases, you can see the price appears in the system before it appears in the market. The system can sometimes order at prices not yet available in the market.

Important Notes

- *The service uses SSI securities JSC's API*. Currently, there are only a few securities companies that provide public APIs for their clients. ALGOTRADE chooses SSI's API because it's easily accessible to the majority of investors.
- *Investors need to establish API connections*. ALGOTRADE LAB is a tool to support investors to perform transactions on their own accounts through API. Investors set up trading parameters based on their requirements/wishes. To ensure safety, investors should only deposit sufficient money in the account to open up to 01 futures contract.
- *The algorithm is not expected to make profits*. The lab uses the SMA algorithm, detailed in article 61. It's simple and accessible and the goal is to help investors have a detailed view of the operation of an automated trading algorithm. Note that SMA isn't guaranteed to make profits. In fact, over the long term, this algorithm on average tends to incur losses.
- *Profits or losses solely belong to investors*. During the lab experience, there will be lucky investors who make profits and vice versa. ALGOTRADE LAB hopes that investors will see this result as a result of luck, not the result of the algorithm.
- *Any asset changes in securities accounts are within the scope of investors' rights and responsibilities*. ALGOTRADE will not have any responsibility and liabilities related to investors' profits or losses.
- *ALGOTRADE LAB uses Python programming language*. We choose Python programming language to provide the algorithmic trading experience for investors. The source code is publicly available. Investors will see detailed information processing with each price tick.

- *The service is completely free*. ALGOTRADE LAB service is completely free, without any benefits from related parties, including SSI Securities JSC.

Why You Should Try Algorithmic Trading Now

In algorithmic trading, real experience is highly valuable compared to just a theoretical approach. In developed countries, only professional traders or professionals working at investment banks and investment funds have the ability to gain experience in the field. Therefore, this is a good opportunity for investors to gain experience in algorithmic trading. Moreover, investors can adjust algorithm parameters to suit investment preferences. This experience is hard to find in trading software like AmiBroker, MetaTrader, or TradingView.

For investors interested in algorithmic trading, the ALGOTRADE LAB experience will help them find new directions to create their own technical analysis and automated trading systems. They no longer have to rely on third-party software.

For other investors, the experience will help investors understand the market in the near future. When trading on the Vietnamese stock market, they will anticipate how most transactions take place, thereby developing a strategy most suitable for their investment preference.

61

INTRODUCTION TO SMA ALGORITHM

Definition of SMA

The moving average (MA) is one of the classic tools used by technical analysts. Simple moving average (SMA) is the most common form of MA, defined as follows:

- Collect past price points data:
  $P_{t-N+1}, ..., P_{t-3}, P_{t-2}, P_{t-1}, P_t$
- Select the number of price points (N) to be averaged;
- Calculate the average of the last N price points:
  $SMA(N)_t = (P_t + P_{(t-1)} + P_{(t-2)} + ... + P_{(t-N+1)}) \div N$

SMA Algorithm

Hypothesis: When the price starts to rise again after a downtrend, the price line crossing over the SMA signals an uptrend that is expected to continue in the short term. Conversely, when the price starts to fall after an uptrend, a drop below the SMA signals the beginning of a downtrend.

Market: Derivatives market – futures contracts VN30F1M.

Strategy Group

☒	Momentum	☐	Event-Driven
☐	Mean Reversion	☐	Market maker
☐	Market Neutral	☐	Scalping
☐	Arbitrage	☐	Smart Beta
☐	Grid Trading	☐	Sniffing
☐	Front-running ETF	☐	Trading Cost Optimization

When to Open Positions

- Independent variable: VN30F1M price by ticks.
- Dependent variable:
  $SMA(N)_t = (P_t + P_{t-1} + P_{t-2} + ... + P_{t-N+1}) \div N$
  Where P_t is the current price of VN30F1M, P_{t-1} is the previous tick price.
- Parameter: the number of tick price points (N) to calculate the average, default = 2000.
- Place a limit order (LO) to buy at the ceiling price or sell at the floor price as soon as there's a signal to open a position:
  If $P_{t-1} < SMA(N)_{t-1}$ and $P_t \geq SMA(N)_t$ then open a buy position;
  If $P_{t-1} > SMA(N)_{t-1}$ and $P_t \leq SMA(N)_t$ then open a sell position.
- Only trade 01 contract maximum for each position. Limit 03 positions maximum per day (03 transactions to open positions and 03 transactions to close positions).

When to Close Positions

- Place orders to sell at the floor price or buy at the ceiling price as soon as the target is reached.

`Take-profit point = Position opening price + Expected profit`

- If P_t ≤ `(Price to open the position – Stop loss)` then place an order to buy at the ceiling price or sell at the floor price to close the position (stop loss). The stop loss threshold can be adjusted.
- No positions remain overnight. At the ATC sessions, if the account still has an open position, then close the position with an ATC order.

Results

- Expected profit: N/A;
- Maximum drawdown (MDD): N/A.

Note

- The SMA algorithm works well when the market has a clear trend.
- SMA algorithm is a popular and simple algorithm in technical analysis, designed to illustrate the concept of "*trading algorithm*" and test the automated system. Therefore, investors should note that this algorithm does not guarantee profits.

62

HOW TO REGISTER FOR API AT SSI SECURITIES JSC

Currently SSI Securities Joint Stock Company provides Fast Connect API service to customers with the following details (updated March 31, 2023):

Fast Connect Data	Fast Connect Trading
■ Real-time market data ■ OHLC daily and historical intraday data ■ Index data, stock details ■ Backtest data	■ User authentication ■ Place/cancel/modify orders ■ Account, portfolio, order book information ■ Streaming order status ■ Streaming derivatives portfolio ■ Derivatives conditional orders

Source: www.ssi.com.vn/khach-hang-ca-nhan/fast-connect-api

How to Sign Up for Fast Connect API

Contact the branch office of SSI Securities JSC to register and subscribe to the service.

How to Create a Connection Key

Visit www.iboard.ssi.com.vn > Support > API services > Click on the “key” icon to generate a connection key > Select “Generate

new connection key" > Click "Continue" > Enter OTP verification code > Click "Confirm".

After completing the steps above, the system will generate the key to save and start using the service.

Figure 28 Example of an API Access Key

At this point, you will have access to the API tool to start a new journey as an algorithmic trader and experience automated trading at ALGOTRADE LAB.

63

EXPERIENCE ALGOTRADE LAB

After opening an account at SSI for the derivatives market, registering for the API service, investors can experience algorithmic trading at ALGOTRADE LAB with a minimum deposit to open 01 futures contract in their account. The high-level steps are as follows:

Open an account > Sign in > Set up API connection > Algorithmic parameter configuration > Algorithm launch > Algorithm monitoring.

Open an Account and Sign In

Submit your account details at www.algotrade.vn/lab > Receive account information by email > Access www.lab.algotrade.vn and login to get started.

The directory structure of the algorithm includes:

- *config.ipynb*: to configure the API and configure the parameters of the SMA algorithm;
- *data.ipynb*: to store tick values of VN30F1M futures contract and compute `SMA(t)`;
- *db.ipynb*: a simple database to store order status, net profit/loss, etc.
- *main.ipynb*: implement the algorithm and review execution log in real-time;
- *logs.log*: contains the SMA value and all order information the algorithm has traded (only to check algorithm accuracy).

Set Up API Connections

At the main interface of ALGOTRADE LAB, open file "config. ipynb" > Enter the required information/desired information in the following lines: `CONSUMER_ID, CONSUMER_SECRET, PRIVATE_KEY, ACCOUNT, OTP`.

```
CONSUMER_ID='changeme'
CONSUMER_SECRET='changeme'
ACCOUNT='changeme'
PRIVATE_KEY='changeme'
OTP='changeme'

HUB_URI='http://1.55.212.27:3000'
CUT_LOSS_THRESHOLD=-3
TAKE_PROFIT_THRESHOLD=3
MAX_ROUND=2 # if greater than 3, it will be set to 3 as default
START_TRADING_TIME='09:00:00' # between 09:00:00 and 14:30:00
```

Algorithm Parameter Configuration

Under the API connection key input, investors can adjust the algorithm parameters to their wishes. The following parameters can be changed:

```
CUT_LOSS_THRESHOLD=-1
TAKE_PROFIT_THRESHOLD=1
MAX_ROUND=2 # if greater than 3, it will be set to 3 as default
START_TRADING_TIME='09:00:00' # between 09:00:00 and 14:30:00
```

- `CUT_LOSS_THRESHOLD`: stop loss parameter. When reaching this threshold, the system will automatically close all open positions with market orders. The stop loss value will be derived from the current price and the actual open price, not the theoretical price. The default parameter is -3.

- `TAKE_PROFIT_THRESHOLD`: take-profit parameter. The default parameter is 3. The system will automatically take profits when reaching this threshold.
- `MAX_ROUND`: the number of times the algorithm can activate. One activation means one open position and one close position. The default parameter is 3, so there are 6 transactions in total. This is to ensure the investor's experience without excessive trading per day.
- `START_TRADING_TIME`: the algorithm start time. Investors can set any time during trading hours from 09:00:00 to 14:30:00. The default is 09:00:00.

After adjusting the configuration, click save (Ctrl + S or Command + S) to update the changes to the system. Here's an example illustrating an investor who has completed the API connection and parameter configuration:

```
CONSUMER_ID='6b661643b4dd485dbb132ce64c09b9d5'
CONSUMER_SECRET='cfa82c0ea7da1a439d2e07ac3c7caeee'
ACCOUNT='1111111'
PRIVATE_KEY='PFJTQUtleVZhbHVlPjxNb2R1bHVzPnVRSXFJVll6dDBtddsfse
FF5ZGh6c0I3cVZQd29Sa1pRT08L01vZHVsdXM+PEV4cG9uZW50PkFRQUI8L0V4'
OTP='123456'
HUB_URI='http://1.55.212.27:3000'
CUT_LOSS_THRESHOLD=-3
TAKE_PROFIT_THRESHOLD=5
MAX_ROUND=2 # if greater than 3, it will be set to 3 as default
START_TRADING_TIME='10:30:00' # between 09:00:00 and 14:30:00
```

Algorithm Launch

After connecting the API and configuring algorithm parameters at the main.ipynb file, select ”Kernel” > click “Restart Kernel and Run All Cells” to start the algorithm.

Algorithm Monitoring

At the ALGOTRADE LAB interface, open "main.ipynb" to observe two key parts:

- *"SSI Algorithm Demo"*: shows the code of the SMA algorithm;
- *"If you want to show log realtime"*: located under "SSI Algorithm Demo", it displays real-time trading algorithm status updates.

After initializing the algorithm, the prompt "*If you want to show log realtime*" will continuously show updated information:

```
INFO:root:Time: 09:55:25, SMA(t-1): 1042.39, SMA(t): 1042.12,
LAST_PX(t-1):1038.7, LAST_PX(t):1038.7
INFO:root:Time: 09:55:25, SMA(t-1): 1042.22, SMA(t): 1042.16,
LAST_PX(t-1):1038.7, LAST_PX(t):1038.7
INFO:root:Time: 09:55:25, SMA(t-1): 1042.16, SMA(t): 1042.81,
LAST_PX(t-1):1038.7, LAST_PX(t):1038.7
INFO:root:Time: 09:55:27, SMA(t-1): 1042.11, SMA(t): 1042.61,
LAST_PX(t-1): 1038.7, LAST_PX(t): 1038.7
INFO:root:Time: 09:55:27, SMA(t-1): 1042.16, SMA(t): 1042.14,
LAST_PX(t-1): 1038.7, LAST_PX(t): 1038.7
```

Stop the Algorithm

In the file main.ipynb, click "Kernel" > "Shut Down Kernel" to stop the algorithm. The status line "KeyboardInterrupt" will display when the algorithm stops successfully.

```
INFO:root:Time: 11:05:55, SMA(t-1): 1038.71, SMA(t): 1038.98,
LAST_PX(t-1): 1037.4, LAST_PX(t): 1037.3
INFO:root:Time: 11:05:55, SMA(t-1): 1038.98, SMA(t): 1038.76,
LAST_PX(t-1): 1037.3, LAST_PX(t): 1037.3

-----
KeyboardInterrupt
Traceback (most recent call last)
Cell In[21], line 1
----> 1 init_printing_log()

Cell In[20], line 12, in init_printing_log()
     10 logfile = open("logs.log", "r")
     11 loglines = tail_log(logfile)
---> 12 for line in loglines:
     13     print(line, end='')

Cell In[20], line 5, in tail_log(thefile)
      3 line = thefile.readline()
      4 if not line or not line.endswith('\n'):
----> 5     time.sleep(0.1)
      6     continue
      7 yield line
```

Adjusting Algorithm Parameters During Operations

If you want to change the algorithm parameters when the system is running, you can do the following:

Open “config.ipynb” > Re-enter new parameters to be changed > Click save (Ctrl + S or Command + S) > In the main interface, click ”Kernel” > click “Restart Kernel and Run All Cells” to restart the algorithm.

Note that it’s not necessary to stop the algorithm when changing its parameters.

Source Code

The following is a detailed explanation of the main source code snippets.

```
import ujson
import time
import logging
from threading import Thread
from algotrade import redis, ssi_api, handler

logging.basicConfig(filename='logs.log', level=logging.INFO)
```

Import necessary libraries to run the algorithm, including the internal ALGOTRADE library.

```
def handle_order_event(msg):
  event_data =  ujson.loads(msg['data'])
  log("event_data['data']: {}".format(event_data))
  data = event_data['data']
  event_type = event_data['type']

  if event_type == 'orderEvent':
    global OPEN_POSITION, CURRENT_ORDER, TOTAL_PROFIT
    global HANDLED_REQUESTS, CANCELLATION_LOCK
    request_id = data['uniqueID']
    log('------------BEFORE UPDATE------------')
    log('OPEN_POSITION {}'.format(OPEN_POSITION))
    log('CURRENT_ORDER {}'.format(CURRENT_ORDER))
    log('-------------------------------------')
    log('HANDLED_REQUESTS: {}'.format(HANDLED_REQUESTS))
    if CURRENT_ORDER['request_id'] == request_id:
      CURRENT_ORDER['order_id'] = data['orderID']
      if (
        data['filledQty'] > 0
        and request_id not in HANDLED_REQUESTS
      ):
        # avoid same order events are returned
        HANDLED_REQUESTS.append(request_id)
```

```
        # update TOTAL_PROFIT
        filled_price = data['avgPrice']
        TOTAL_PROFIT += calculate_profit(filled_price)
        log('TOTAL_PROFIT {}'.format(TOTAL_PROFIT))

        # update OPEN_POSITION
        OPEN_POSITION['avg_price'] = filled_price
        new_qty = (
          abs(OPEN_POSITION['qty'] - data['filledQty'])
        )
        OPEN_POSITION['qty'] = new_qty
        if new_qty > 0:
          openned_side = CURRENT_ORDER['side']
        else:
           openned_side = None
        OPEN_POSITION['side'] = openned_side

         # update CURRENT_ORDER
         CURRENT_ORDER['status'] = 'FILLED'

         log('----------AFTER FILLED----------------')
         log('----------AFTER FILLED----------------')
         log('OPEN_POSITION {}'.format(OPEN_POSITION))
         log('CURRENT_ORDER {}'.format(CURRENT_ORDER))
         log('--------------------------------------')

       update_redis_db()
```

Process position data from SSI, update the opened positions as well as the total profit/loss when the positions are closed. Then update the status of `OPEN_POSITION` to the initial state (None) to be able to open new positions if the algorithm spots new signals.

```
%run data.ipynb
%run db.ipynb
%run config.ipynb
```

```
r = redis.init_redis()
pubsub = r.pubsub()
F1 =  redis.get_key('F1M_CODE')
F1_TICK_CHANNEL = 'HNXDS:{}'.format(F1)
SSI_EVENTS_CHANNEL = 'SSI_{}_EVENTS'.format(ACCOUNT)
pubsub.subscribe(F1_TICK_CHANNEL, SSI_EVENTS_CHANNEL)
```

```
all_f1_data = redis.get_key(ALL_F1M_PRICE_TICKS)
if all_f1_data is not None:
    init_ticks(ujson.loads(all_f1_data))
```

Initialize pub/sub redis to receive processed tick data as well as order status placed by the algorithm.

```
def open_position(side: str, order_type: str, price: float,
    current_order_status = CURRENT_ORDER['status']
    handler.open_position(
        ACCOUNT,
        F1,
        side,
        order_type,
        price,
        reverse,
        MAX_ROUND,
        current_side,
        current_order_status,
        update_current_order
    )
```

This is the function to open a new position with key information: `ACCOUNT` is the user account; F1 is the futures contract VN30F1M; side is '`BUY`' or '`SELL`'; and price is the price to place orders.

```
def handle_position_with_price(side: str, order_type: str,
data: dict, reverse: bool):
    price = CEILING_PRICE if side == 'BUY' else FLOOR_PRICE
    open_position(side, order_type, price, reverse)
```

```
TRIGGER_ATC_TIME = '14:29:50'
def handle_msg_internal(hidden_info: dict):
  last_px = hidden_info['LastPrice']
  trade_time = hidden_info['Time']
  global OPEN_POSITION, START_TRADING_TIME
  global FLOOR_PRICE, CEILING_PRICE
  if last_px is not None:
    [prev_last_px, prev_sma, sma] = add_tick(last_px)
    log(
      'Time: {}, SMA(t-1): {}, SMA(t): {},'
      'LAST_PX(t-1): {}, LAST_PX(t): {}'.format(
        trade_time, get_sma_value(prev_sma),
        get_sma_value(sma), prev_last_px, last_px
      )
    )
    if FLOOR_PRICE is None or CEILING_PRICE is None:
      FLOOR_PRICE = hidden_info['Floor']
      CEILING_PRICE = hidden_info['Ceiling']

    if (
      trade_time >= START_TRADING_TIME
      and trade_time < TRIGGER_ATC_TIME
    ):
      if OPEN_POSITION['side'] is not None:
        unrealized = calculate_profit(last_px)
        log(
          'Unrealized Profit/Loss: {}'.format(
            calculate_profit(last_px)
          )
        )
        if (
          unrealized <= CUT_LOSS_THRESHOLD
          or unrealized >= TAKE_PROFIT_THRESHOLD
        ):
```

```
            # cut loss or take profit -> close opening position
            print('Cut loss' if unrealized < 0 else 'Take profit')
            handle_position_with_price(
              get_reverse_side(OPEN_POSITION['side']), 'LO',False
            )
        elif prev_sma > 0.0:
          if prev_last_px < prev_sma and last_px >= sma:
            log('Long Signal')
            handle_position_with_price('BUY', 'LO', False)
          if prev_last_px > prev_sma and last_px <= sma:
            log('Short Signal')
            handle_position_with_price('SELL', 'LO', False)

    if (
      trade_time >= TRIGGER_ATC_TIME
      and OPEN_POSITION['side'] is not None
    ):
      # close opening position in ATC session
      log('Close ATC')
      handle_position_with_price(
        get_reverse_side(OPEN_POSITION['side']), 'ATC', False
      )
```

This is the main part of the algorithm. After receiving one new tick, the system will compute `SMA(t)` and `SMA(t−1)` and check the condition:

- If `Price(t−1) < SMA(t−1)` and `Price(t) ≥ SMA(t)` then the system will open a new long position;
- If `Price(t−1) > SMA(t−1)` và `Price(t) ≤ SMA(t)` then the system will open a new short position.

If the unrealized profit/loss (for the opened position) exceeds the profit/loss thresholds defined in the file "config.ipynb", then an opposite position will be opened to take profits or stop loss with a limit order at the ceiling or floor price.

At 14:29:50, if the position is still open then the system will open an opposite position to close it at ATC.

64

SMA ALGORITHM CONFIGURATION AND MONITORING EXPERIENCE

Parameter Recommendations

The SMA algorithm is used to test the algorithmic trading system, not to seek profit. The fact has shown that, in the long run, this algorithm tends to make losses. To avoid big losses during the first algorithmic trading experience, it's recommended to set a cut-loss threshold as follows:

- If you choose a stop-loss at `CUT_LOSS_THRESHOLD = –X`
- Then the take-profit threshold should be `TAKE_PROFIT_THRESHOLD = X + Compensation for taxes, fees, & slippage`

The compensation for taxes, fees, and slippage is added to take sufficient profit to cover the trading expenses.

For example, if you choose `CUT_LOSS_THRESHOLD` to -3 points; the taxes, fees, and slippage is approximately 1 point; then `TAKE_PROFIT_THRESHOLD` should be 4 points.

Note, the added compensation value should not be too large; since it's more likely the position will reach stop-loss threshold before having a chance to take any profit.

For `MAX_ROUND` (the maximum number of open positions), it's recommended to set the max value to 3 to get a feeling of the algorithm. When the algorithm opens a buy position and makes a profit, the next new position will be a sell position (the opposite to the

previous one). In contrast, if it realizes a loss, the next new position will be a buy position (the same position to the previous one).

For `START_TRADING_TIME`, it's a simple but important parameter. The algorithm is most effective during periods of markets with clear and continuous up trends or down trends. Therefore, the right start time will greatly contribute to trading performance. For example, suppose investors observe that between 10:00:00 and 11:00:00, the market mostly maintains sideways trends. Then they can choose to activate the algorithm outside of this period to increase the profit likelihood.

Trading System Monitoring

During the system's operation, investors can see the status updates continuously. For the best experience, investors can open the ALGOTRADE LAB interface together with SSI's live board interface to monitor and compare information.

Upon a successful start, stock price and SMA algorithm information will be updated in real time:

```
INFO:root:Time: 09:55:27, SMA(t-1): 1042.11, SMA(t): 1042.61,
LAST_PX(t-1): 1038.7, LAST_PX(t): 1038.7
INFO:root:Time: 09:55:27, SMA(t-1): 1042.61, SMA(t): 1042.42,
LAST_PX(t-1): 1038.7, LAST_PX(t): 1038.7
INFO:root:Time: 09:55:26, SMA(t-1): 1042.12, SMA(t): 1042.22,
LAST_PX(t-1): 1038.7, LAST_PX(t): 1038.7
INFO:root:Time: 09:55:26, SMA(t-1): 1042.12, SMA(t): 1042.13,
LAST_PX(t-1): 1038.7, LAST_PX(t): 1038.6
INFO:root:Time: 09:55:26, SMA(t-1): 1042.13, SMA(t): 1042.83,
LAST_PX(t-1): 1038.6, LAST_PX(t): 1038.6
```

`LAST_PX(t)` is the current price of VN30F1M and `LAST_PX(t–1)` is the previous one. The corresponding SMAs are `SMA(t)` và `SMA(t–1)`.

When reaching the signal to open a position, the trading order is sent and the system updates the status line:

```
INFO:root:Time: 10:56:27, SMA(t–1): 1039.94, SMA(t): 1039.86,
LAST_PX(t–1): 1039.7, LAST_PX(t): 1040
INFO:root:Long Signal
INFO:root:------------BEFORE CALLBACK-----------
INFO:root:CURRENT_ORDER {'order_id': None, 'request_id': None,
'side': '', 'qty': 0, 'price': 0.0, 'status': None}
INFO:root:----------------------------------------
INFO:root:------------AFTER CALLBACK------------
INFO:root:CURRENT_ORDER {'request_id': '20589519', 'order_id':
None, 'side': 'BUY', 'qty': 1, 'price': 1124.2, 'status':
'PENDING_NEW'}
```

In this example, `LAST_PX(t)` is lower than `SMA(t)` and is trending up; when `LAST_PX(t)` exceeds `SMA(t)`, the system automatically sends a buy order.

The system continuously updates the profit/loss status when opening a position:

```
INFO:root:Time: 10:57:22, SMA(t–1): 1039.89, SMA(t): 1039.83,
LAST_PX(t–1): 1039.9, LAST_PX(t): 1039.9
INFO:root:Unrealized Profit/Loss: –0.1
INFO:root:Time: 10:57:25, SMA(t–1): 1039.83, SMA(t): 1039.78,
LAST_PX(t–1): 1039.9, LAST_PX(t): 1039.9
INFO:root:Unrealized Profit/Loss: –0.1
INFO:root:Time: 10:57:25, SMA(t–1): 1039.78, SMA(t): 1039.72,
LAST_PX(t–1): 1039.9, LAST_PX(t): 1039.8
INFO:root:Unrealized Profit/Loss: –0.2
```

`Unrealized Profit/Loss = 0.1` means the current profit is at 0.1 point, equivalent to 10,000 VND profit in a securities account.

When the position reaches the stop-loss threshold, the system will show the following status upon automatically closing the position:

```
INFO:root:Unrealized Profit/Loss: -3.0
Cut loss
INFO:root:------------BEFORE CALLBACK-----------
INFO:root:CURRENT_ORDER {'request_id': '20589519', 'order_id':
'16362499', 'side': 'BUY', 'qty': 1, 'price': 1124.2,
'status': 'FILLED'}
INFO:root:----------------------------------------
INFO:root:------------AFTER CALLBACK------------
INFO:root:CURRENT_ORDER {'request_id': '20749368', 'order_id':
None, 'side': 'SELL', 'qty': 1, 'price': 977.2, 'status':
'PENDING_NEW'}
```

Similarly when the position reaches the take-profit threshold and gets closed, the system will show the following status:

```
INFO:root:Unrealized Profit/Loss: 5.6
Result 20749367 {'message': 'Success', 'status': 200, 'data':
{'requestID': '20749367', 'requestData': {'instrumentID':
'VN30F2303', 'market': 'VNFE', 'buySell': 'B', 'orderType':
'LO', 'channelID': 'TA', 'price': 1124.2, 'quantity': 1,
'account': '1838358', 'stopOrder': False, 'stopPrice': 0.0,
'stopType': '', 'stopStep': 0.0, 'lossStep': 0.0, 'profit
Step': 0.0}}}
Take profit
openned: 2, max_openned: 6
INFO:root:------------BEFORE CALLBACK-----------
INFO:root:CURRENT_ORDER {'request_id': '16464644', 'order_id':
'16361124', 'side': 'SELL', 'qty': 1, 'price': 977.2,
'status': 'FILLED'}
INFO:root:----------------------------------------
INFO:root:------------AFTER CALLBACK------------
INFO:root:CURRENT_ORDER {'request_id': '20749367', 'order_id':
None, 'side': 'BUY', 'qty': 1, 'price': 1124.2, 'status':
'PENDING_NEW'}
```

To avoid technical errors, do not manually trade while the algorithmic trading system is operating.

However, the system can encounter unexpected bugs and are unable to close positions, despite reaching the take-profit or stop-loss threshold. In this case, it's important to stop the system and place a manual order to close the positions and avoid unexpected consequences.

GLOSSARY

Indicator Identifies the ticker symbol and the appropriate time to trade. Indicators are divided into technical indicators such as Moving Average (MA) and Relative Strength Index (RSI), and fundamental indicators such as Price-to-Earning Ratio (PE) and Return-on-Equity (ROE). Depending on the algorithms, an indicator can be defined based on its intended use.

Parameter (Algorithm) A predefined arithmetic value like price, volume, or parametric constants of mathematical models.

Trading logic A rule-based system used to make trading decisions based on indicators and parameters.

Trading algorithm A set of statements and trading logic to execute trading decisions, including but not limited to buy or sell orders and price, volume, and order types. Within the scope of our book, "algorithms" without further explanation can be understood as "trading algorithms".

Algorithm trading The practice of using a computer system to carry out fully automated trades according to pre-programmed algorithms.

Algorithmic trading system An automated system with data collection processes and data queries. The system uses computer algorithms to make trading decisions and reports, to manage real-time financial portfolios without any human intervention.

Algorithm hypothesis An assumption with a solid financial foundation that may yield long-term profits.

Algorithmic trader An investor or trader that uses an algorithmic trading system.

Semi-automated trading A system that uses computers combined with human intervention to optimize the decision-making process.

In-sample historical data The past historical data used to train and optimize trading algorithms.

Out-of-sample historical data The past historical data used to validate trading algorithms after the training phase. It is to assess the profitability of an algorithm in the future. In-sample and out-of-sample historical data do not have any overlap.

Future data (Unseen data) Out-of-sample data and different from past data.

Backtesting The use of past data to evaluate an algorithm's performance.

Forward testing The use of future data from the present to a certain point in the future to evaluate an algorithm's performance.

Paper trading A stage in testing future data. It uses real-time data in a simulated trading environment to evaluate an algorithm's performance.

Optimization A process in algorithmic trading to find the parameter values for a trading algorithm to produce the best performance during live trading in the target market.

Overfitting A phenomenon in algorithmic trading that the post-optimization algorithm gives good results on in-sample historical data only.

API (Application programming interface) A set of definitions and protocols that allow communication between two software applications.

High-frequency trading An algorithmic trading strategy with an extremely large number of transactions at high speed. It opens and closes positions in a very short period of time.

Price momentum strategy Investors buy rising stocks in terms of price (or short-sell falling stocks) with the reasoning that stocks will follow their momentum trends in price.

Mean-reversion strategy A strategy to buy stocks when the price is lower than their intrinsic or average value; and vice versa sell or short when the price is higher.

Arbitrage trading strategy A strategy to take advantage of temporary price differences for the same asset in two different markets. This trading strategy makes profits without much risk.

Market-neutral strategy A group of strategies when investors open both long and short positions to minimize the effects of market risk on portfolio profitability.

Pair trading A strategy that finds two stocks with high correlation to match a long position with a short position. When their correlation deviates above the long-term average, investors match a long position on the underperforming stock, and a short position on the outperforming stock, expecting the deviation to be temporary. When the price correlation converges to the average, the investor closes the position to realize the profit.

Statistical arbitrage strategy A group of market-neutral strategies developed from the pair trading strategy. It uses statistical mathematical models with computer systems to identify trading opportunities. They are mostly from unusual relative price changes from one stock compared to another.

Event-driven trading strategy A trading strategy that takes advantage of market inefficiencies from corporate events like mergers and acquisitions, corporate restructures, share buybacks, extraordinary dividends, etc. to trade in the short term.

Smart-beta strategy Also known as a factor-based strategy, a strategy to build portfolios according to rule-based processes. It uses business factors like liquidity, value, and quality as criteria for making trading decisions.

Scalping strategy A special strategy that focuses on an ultra-short time frame to open and close positions in order to make many thin profits.

Front-running ETF strategy Investors anticipate the action of exchange-traded funds (ETFs) according to the public prospectus and simulate part of the action prior to the fund's rebalancing.

Market-making strategy A strategy of simultaneously placing orders at the best bid price and the best ask price. This strategy makes profits from the bid-ask spread.

Grid trading strategy A strategy that sets up a price grid around a predefined value to make profits from market fluctuations independent of market trends.

VWAP algorithm Breaks down and places orders at different times and different volumes based on historical data of trading volume.

TWAP algorithm Breaks down the order into equal parts and consistently places orders, separated by an equal amount of time.

POV algorithm Breaks down the order with volume based on the percentage of volume according to the market liquidity. As the market trading volume increases, it will trade more stocks and vice versa.

Implementation shortfall A discrepancy between paper profits and actual profits. Paper profit is a theoretical profit under the assumption all orders are executed at the desired price and volume without any fees.

TWAP (Time-weighted average price) the equally weighted average price of all orders made during the execution period.

VWAP (Volume-weighted average price) the weighted average price by the volume of all orders made during the execution period.

Maximum drawdown (MDD) The maximum loss from the peak to the bottom for an asset or portfolio.

Sharpe ratio The rate of expected return minus the risk-free rate, divided by the standard deviation of the rate of return. It measures the expected performance per unit of risk.

Kelly criterion Determines the optimal capital per trade to maximize the long-term performance of a trading algorithm.

ALGORITHMIC TRADING THEORY AND PRACTICE
A Practical Guide With Applications on the Vietnamese Stock Market

THE GIOI PUBLISHERS
Headquarters:
No. 46 Tran Hung Dao, Hang Bai Ward, Hoan Kiem District, Ha Noi
Tel: 0084.24.38253841
Office:
No. 7 Nguyen Thi Minh Khai, Ben Nghe Ward, District 1, Ho Chi Minh City
Tel: 0084.24.38220102
Email: thegioi@thegioipublishers.vn
Website: www.thegioipublishers.vn

Printed in 500 copies, at 14.5 x 20.5 cm by Thuong Mai Quang Cao & In Phu Sy, LLC
Address: P407 A10 Khuong Thuong, Trung Tu, Dong Da, Ha Noi

Publication registration number: 2457-2023/CXBIPH/11-138/ThG

Publication approval number: 989/QĐ-ThG on August 03, 2023

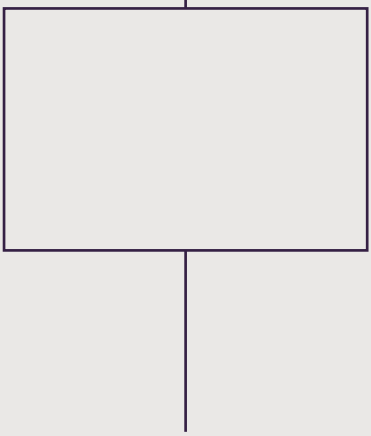
Published by: Pham Tran Long
Director and Chief-Editor of The Gioi Publishers

ISBN: 978-604-77-7529-3
Printed and registration of copyright in 2023

Editor: Trinh Hong Hanh

Cover and layout design:
Pham Tuan Tai

Distributed by: DIMI Digital, LLC
Address:
87 Le Hong Phong, Hai Chau District, Da Nang City



Algorithms execute over 50% of total trading volume in the global market. In Vietnam, algorithmic trading is still at the very beginning.

In this book, we hope to share the experiences, risks, and potentials of algorithmic trading when applied in the Vietnamese market.

Do you believe algorithmic trading will profoundly change your investment decisions and trade executions?

Will you become an early bird in this wave or will you just keep on waiting?

DIMI BOOK
We deliver knowledge and happiness

ONLINE VERSION

ALGOTRADE LAB

ISBN: 978-604-77-7529-3

9 786047 775293

Price: 721.000 ₫