Thesis

Neo

Abstract

things to put in abstract

1 Intorduction

People didn't like to invest in the past because they weren't familiar with investment. What they do with their income is to deposit it in the bank, which seems neither a good nor a bad choice at the moment. As the time goes by, the unseen problem grows day by day, that is the inflation. As people realize the problem with depositing money in the bank, more and more people are willingly to invest to gain their wealth other than keeping the money in the bank. Hence, investment becomes an important issue in today's society.

There are varieties of investment to choose from, such as, stocks, bonds, options, real estate, and futures. One of a popular investment target is stocks. The advantages of investing in stocks are transparency, liquidity, and low requirement to invest.

The U.S. stock market is the target in this research. It is the largest economy in the world which has a great impact on the world financial system. Any oscillation in the U.S. stock market will reflect on every stock market all over the globe. Also there are a lot of companies and various types of companies from different country in the U.S. stock markets. Hence, the diversity and market capitalization make it a good choice of investing in the U.S. market.

The basic idea of investing in stocks is to buy at low price and sell at high price. The idea is simple, but hard to achieve without any analysis of the stocks market. So, most people use two types of methods to analyze the company or the stock markets that investors are interested in. There are two methods for investors to analyze whether a stock is worth buying or not, one is fundamental analysis, the other one is technical analysis.

Fundamental analysis in the stock market is a method of evaluating the operating conditions of a company, such as governance, revenues, earnings, future growth, return on equity, profit margins, and other data. All of this data is available in a company's financial statements. These factors refer to estimating a company's underlying value and potential for future growth. Buy and sell decisions are then made based on whether the investors believe that those factors will have positive or negative impact on the stock price. If an analyst calculates that the value of the stock is expected to be significantly higher than the current market price of the stock, investors may want to buy shares. If the analyst calculates a lower intrinsic value than the current market price, the stock is considered overvalued and investors may want to sell the stock.

Technical analysis is very different from fundamental analysis. It is an umbrella term for using mathematical method to evaluate stock prices. The objective of using technical analysis is not to predict the price, it gives the investors the signal of when to buy or sell a stock by evaluating the statistical trends of stock price. Because it is believed that history tends to repeat itself, any oscillation shall finally be reflected in the stock price. Technical analysis is often used as a short-term trading strategy. Technical analysis is used generally to evaluate the price changes, but some analysts track numbers of data other than just price, such as trading volume or open interest figures.

There are hundreds of patterns and signals that a company can generate. Financial researchers have been studying these signals to support technical analysis trading. Various technical indicators have also been developed to help them better understand the behavior of stock markets.

In this research, we use technical analysis rather than fundamental analysis. The reason is that the data needed for technical analysis are easier to access, which is the stock price. Besides, it is also easier to focuse only on the stock price, not on several statistic data like governance, revenues, earnings, etc.

The technical indicators we use are relative strength index (RSI) and simple moving average (SMA). These two indicators are the most popular among the investors. There are several parameters that should be given to the indicators before actually using them. These parameters are important because they decide when the investors should enter or exit the stock market.

Finding a good set of parameters becomes a great issue here because there are vast number of combinations of the parameters. Finding a good set of parameters to use is a complex problem. Using the exhaustive method is probably not a smart idea because it takes too much time and resources. Metaheuristic algorithm is capable of coping with this optimization task. There are many metaheuristic algorithms, such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and Global Quantum-inspired Tabu Search algorithm with Notgate (GNQTS) [A Novel Portfolio Optimization with Short Selling Using GNQTS and Trend Ratio], etc. The method that we apply is GNQTS which can find parameters efficiently by using global-best as guidance and quantum not gate to escape local optima. Additionally, our research proposed new sliding windows of different time frames to maximize the profit. Moreover, we combine RSI and SMA to form better strategies rather than using a single technical indicator.

In brief, our research applies two technical indicators with GNQTS and sliding windows. By Utilizing GNQTS to optimize parameters for both RSI and SMA and apply them in the U.S. stock market. The results of experiment show that our method is encouraging and is head and shoulder above the buy and hold strategy (B&H) and traditional strategies.

2 Related Work

Research on stock markets has attracted many scholars' attention in different fields. There are three commom methods of analyzing the market, metaheuristic, artificial neural network, and fuzzy theory. Our research focus on using evolutionary computation to manage the problem.

In a stock trading problem, fuzzy theory is used to categorize a huge amount of historical stock price data by a fuzzy classification. The aim is to predict future stock movements.

This paper [Embedding Four Medium-Term Technical Indicators to an Intelligent Stock Trading Fuzzy System for Predicting: A Portfolio Management Approach] utilizes a small number of coherent trend-following technical indicators with similar characteristics, but constructed with a different philosophy, in order to predict the movement of a stock market.

[A Hybrid Artificial Neural Network with Metaheuristic Algorithms for Predicting Stock Price] aims to predict prices on stock exchange via the hybrid artificial neural network models and metaheuristic algorithms which consist of cuckoo search, improved cuckoo search, improved cuckoo search genetic algorithm, genetic algorithm, and particle swarm optimization. The results suggest that particle swarm optimization is a dominant metaheuristic approach to predict stock price according to statistical performances of the above approaches.

[NSE Stock Market Prediction Using Deep-Learning Models] used four deep learning architectures for prediction of NSE and NYSE. From the result, deep learning models is capable of capturing the abrupt changes in the system.

[Technical analysis strategy optimization using a machine learning approach in stock market indices] propose a hybrid approach to generate trading signals. The method consists of applying a technical indicator combined with a machine learning approach to produce a trading decision. The performances of Linear Model (LM), Artificial Neural Network (ANN), Random Forests (RF) and Support Vector Regression (SVR) was tested. Results achieved show that the addition of machine learning techniques to technical analysis strategies improves the trading signals.

In this research [Technical Market Indicators Optimization using Evolutionary Algorithms], technical indicators are applied to interpret stock market trending and investing decision. In this work, Evolutionary Algorithms are proposed to discover correct indicator parameters in trading. In order to check this proposal the Moving Average ConvergenceDivergence (MACD) technical indicator has been selected. Preliminary results show that this technique could work well on stock index trending. Indexes are smoother and easier to predict than stock prices.

Evolutionary computation is a group of algorithms inspired by biological evolution for global optimization. The process of evolutionary computation can be divided into several major steps such as reproduction, mutation, recombination, natural selection and survival of the fittest. It is a promising technology that have been used in science and engineering for solving practical problems and as computational models. The characteristic of evolutionary computation is when searching the solution within the solution space, the solution algorithm finds gets closer and closer to the best solution each iteration, because of the experience from the last iteration. There are few popular evolutionary algorithms such as GA, ACO and PSO.

GNQTS is based on the Quantum-inspired Tabu Search algorithm (QTS) [Classical and quantum-inspired Tabu search for solving 0/1 knapsack problem], which takes the advantages of the classical Tabu search and the characteristics of quantum superposition. QTS outperforms other heuristic algorithms whn it comes to optimization problems. It uses the best solution and the worst solution in each iteration as guidance to update the probability of choosing which item to put into the knapsack. As the result of the QTS, the particles approaches the best solution in the solution space in each iteration while moves away from the worst solution simultaneously. GNQTS [A Novel Portfolio Optimization with Short Selling Using GNQTS and Trend Ratio] keeps the advantages of QTS and adding the quantum Not-gate to enhance the ability in order to leave the local optimal. GNQTS has improved performance and stability when looking for potential solutions.

3 Proposed Method

3.1 Technical Indicators

Technical indicators are the rule of thumb or pattern-based signals produced mathematically by the stock price or volume. The foundation of technical indicators is the historical prices of the stocks. It is believed that history will repeat itself as time extends. In other words, patterns of market behavior continuously appear throughout the history of the stock market. By analyzing the historical data, technical analysis uses indicators to determine the timing to buy or sell stocks.

3.1.1 Moving Average (MA)

A Moving Average is an indicator that shows the trend of stock price of a company. If the moving average was decreasing, it indicates that the price is falling recently. If the moving average was increasing, it indicates that the price is rising recently. There are several different types of moving averages. The most popular one is the Simple Moving Average (SMA), which is the indicator that is used in this research. The main difference between the moving averages is that the weighting applies to the price of stocks when calculating the indicator.

SMA is the average closed price of a certain period of time (e.g., 5 days). The period of days that is been used to calculate the average price is called the look-back period. Among all the MA, SMA is an indicator that can be easily calculated, because the weight, which applies to the price of stocks when calculating SMA is equally weighted. The formula of SMA is shown in 1, where N is the look-back period and T is the date of today.

$$SMA_N = \frac{price_{T-N} + price_{T-N+1} + price_{T-N+2} + ... + price_{T-2} + price_{T-1}}{N} \tag{1}$$

The most common way to use MA is to compare the relationship between two MA trends, known as crossover. The way to define a crossover is that when plotting two different MA values, the first MA line crosses through the second MA line from the bottom. This is also referred to as a golden cross. On the other hand, a death cross is when the first MA line crosses through the second MA line from above. We can simplify the trading strategy of using these two MA into MA(MA_1, MA_2). Table 1 shows the parameters of traditional MA that are frequently used by investors. The combination of the traditional strategy is restricted. Only two types of strategies are allowed, MA(Short-term, Mid-term), MA(Mid-term, Long-term). Hence there are 8 strategies in traditional MA.

Table 1: Tradition MA strategies

Short-term	**Mid-term**	**Long-term**
5 days (one week)	20 days (one month)	120 days (half year)
10 days (two weeks)	60 days (three months)	240 days (one year)

Figure 1 demonstrates the timing of golden cross and death cross when using SMA(5, 20). A buy signal is triggered when a golden cross appears. A selling signal is triggered when a death cross appears. These two types of crossover are the important signal to determine the timing of buying or selling the stocks.

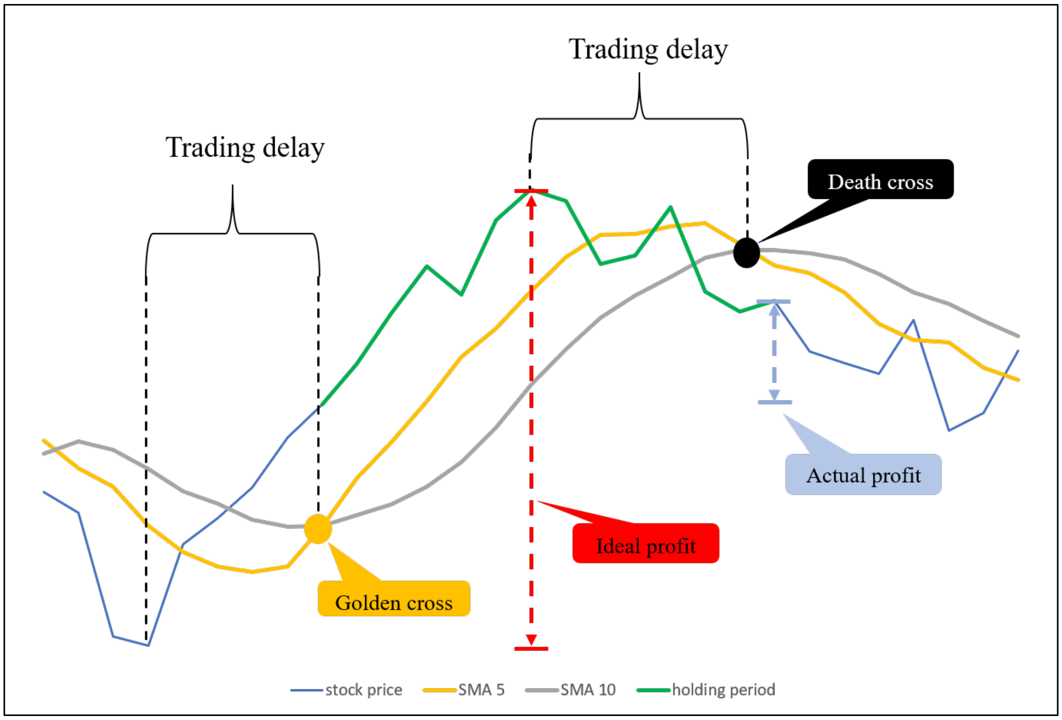

Figure 1: Demonstration of using strategy SMA(5, 20)

Even though MA is a popular investment indicator, there are still some downsides. The first issue is that the moving average is a lagging indicator. As we can see in figure 1, the golden cross and death cross are lagging behind the best time to buy or sell shares, leading to lower profits. The second issue is that there are too few traditional strategies. It is difficult to make a profit with just those few strategies. The third problem has to do with the lack of parameters. MA uses just two parameters to find the golden cross and the death cross. Using the same parameters to buy and sell does not appear to be sufficient.

In order to break the boundaries of traditional MA strategy and find more accurate buying and selling points, this study extends the parameters of MA. There will be four parameters, two for buying and two for selling, rather than jus two for buying and selling. In addition, the parameters can be selected from 1 to 256, instead of choosing from short-term, mid-term and long-term. This significantly increases the number of strategies from 8 to 256^4. More than 4.2 billion strategies that can be selected when using MA as an indicator. Comparison of MA traditional strategy and new MA strategy are shown in table 2.

Table 2: Comparison of SMA traditional strategy and new MA strategy

	traditional strategy	**new MA strategy**
strategy	MA(MA_1, MA_2)	MA(MA_{buy_1}, MA_{buy_2}, MA_{sell_1}, MA_{sell_2})
solution space	8	2^{32}

3.1.2 Relative Strength Index (RSI)

Relative Strength Index (RSI) is a momentum oscillator that was first introduced by J. Welles Wilder, Jr. [3] in 1978. This is a popular indicator in financial technical analysis that measures the magnitude of recent price changes. The basic idea of RSI is to measure how quickly traders are bidding the price of the stocks up or down. RSI sees the uprising stocks as a buyer's strength, and the downswing stocks as a sellers' strength, which are referred to as overbought and oversold.

The period of days needed to calculate RSI is referred to as look-back period. The RSI calculation process can be split into two steps. For step one (as shown in formula 2), the average gain and loss is the average upward change and downward change of the price during the look-back period. As for step two, with the result from step one, we can calculate the next RSI using formula 3 recursively, where N is the look-back period. RSI oscillates between 0 and 100.

$$RSI_{step\ one} = 100 - \left[\frac{100}{1 + \frac{Average\ gain}{Average\ loss}} \right] \tag{2}$$

$$RSI_{step\ two} = 100 - \left[\frac{100}{1 + \frac{Previous\ Average\ Gain \times (N-1) + Current\ Gain}{-(Previous\ Average\ Loss) \times (N-1) + Current\ Loss}} \right] \tag{3}$$

When using RSI, there are three parameters: look-back period, overbought, and oversold. First, select the period of days to calculate RSI. Second, choose the thresholds for oversold and overbought, which means that when RSI meets these thresholds, it will trigger the buy or sell signal. The representation of a RSI strategy is RSI ($period$) ($oversold,\ overbought$). Traditionally, RSI is mostly used on a 14-day timeframe for look-back period according to Wilder and is considered oversold when below 30 and overbought when above 70. Most of the investors use 6, 5, or 14 as a look-back period, and oversold and overbought are (30, 70), (20, 80), where the sum of these two parameters is equal to 100. The frequently used parameters of traditional strategies are shown in table 3.

Table 3: Tradition RSI strategies

period	**(oversold, overbought)**
5	(20, 80)
6	
14	(30, 70)

The traditional strategies restrict the probability to find the perfect time to buy or sell stocks. However, this research extends the look-back periods from 1 to 256, oversold and overbought from 0 to 100, and the sum of oversold and overbought does not necessarily need to be equal to 100. For example, RSI(100)(47, 89) or RSI(9)(19, 55), etc. After removing the restriction of traditional strategies, the potential of RSI has been unlocked. There is a great chance to find a strategy which can better adapt to the ever changing stock markets. The comparison of RSI traditional strategy and new RSI strategy is chown in table 4.

Table 4: Comparison of RSI traditional strategy and new RSI strategy

	traditional strategy	**new RSI strategy**
strategy	RSI (5 or 6 or 14) and (20, 80) or (30, 70)	RSI (1 to 256) (0 to 127, 0 to 127)
solution space	6	2^{22}

3.2 Global-best guided Quantum-inspired Tabu Search Algorithm with Quantum Not Gate (GNQTS)

Global-best guided quantum-inspired tabu search algorithm with quantum not gate (GNQTS) is a metaheuristic algorithm inspired by the superposition state of a quantum. There are three main features about GNQTS. First, after each generation, the particles will get closer and closer to the best solution. Meanwhile, keep the particles away from the wrong solutions. Second, the ability of convergence is enhanced by using the global best as a guidance. Third, quantum not gate is the key to escape local optima. With these features, GNQTS is capable of finding good solutions effectively. Figure 2 shows the flowchart of GNQTS. Algorithm 1 is the pseudo code of GNQTS.

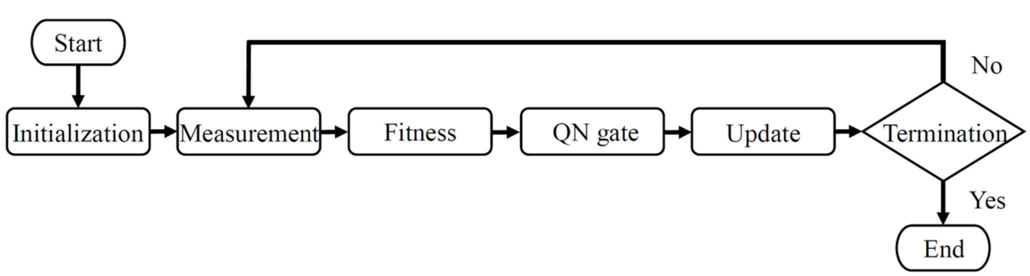

Figure 2: The flowchart of GNQTS

Algorithm 1 GNQTS

$i \leftarrow 0$
Initialize quantum population $Q(0)$
Initialize best solution b
while not termination-condition **do**
    $i \leftarrow i + 1$
    Produce neighborhood set N by measure $Q(i-1)$
    Evaluate $f(s)$
    Find the best solution s^b and the worst solution s^w
    Update b
    Detect whether GNQTS is stuck in local optimal
    if stuck **then**
        Do Quantum Not Gate
    end if
    Update $Q(i)$
end while

3.2.1 Initialztion

In order to explore the potential of RSI and SMA, this article extends the boundary of parameters of these two indicators. We set the look-back period of RSI from 1 to 256, oversold and overbought from 0 to 100. To encode the those bits, we prepare 8 bits for the look-back period, 7 bits for the oversold and the overbought, so there are 22 bits in total. The same rule applies to SMA as well, set the look-back period of 4 parameters from 1 to 256, 8 bits for each of them, so there are 32 bits in total. After determining how many bits of RSI and SMA, we use RSI as an example the following steps.

At the beginning of the algorithm, the probability of choosing each bit is stored in a array called beta matrix. Each bit in the beta matrix is set to 0.5, as in superposition state of quantum - that is, the probability of choosing 0 or 1 is 50%. As shown in 3, where $0 \le n \le 21$.

···	7	6	5	4	3	2	1	0	*n*
···	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	*beta matrix*

Figure 3: Initilize beta matrix

3.2.2 Measurement

After initializing the beta matrix, a random number r is given to each bit to determine the bit should be 0 or 1, where $0 \le r \le 1$. Then we compare each bit with its r, if the probability of bit_n in the beta matrix is greater than r, bit_n is set to 1, else set to 0. The process of measurement is shown in 4.

···	7	6	5	4	3	2	1	0	*n*
···	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	*beta matrix*
	$\wedge$	$\wedge$	$\wedge$	$\wedge$	$\vee$	$\vee$	$\wedge$	$\vee$	
···	0.65	0.79	0.63	0.98	0.22	0.46	0.55	0.37	*r*
···	0	0	0	0	1	1	0	1	*bit*

Figure 4: The process of forming a trading strategy in measurement

After each bit is measured, we can transform these bits into decimals to form a trading strategy as shown in figur 5. In the example, 8 bits for RSI look-back period, 7 bits for the oversold and overbought. The RSI look-back period starts from 1, so add 1 to it.

bit $_7$	*bit* $_6$	*bit* $_5$	*bit* $_4$	*bit* $_3$	*bit* $_2$	*bit* $_1$	*bit* $_0$	14	*RSI look back period*
0	0	0	0	1	1	0	1		
	bit $_{14}$	*bit* $_{13}$	*bit* $_{12}$	*bit* $_{11}$	*bit* $_{10}$	*bit* $_9$	*bit* $_8$	20	*oversold*
	0	0	1	0	1	0	0		
	bit $_{21}$	*bit* $_{20}$	*bit* $_{19}$	*bit* $_{18}$	*bit* $_{17}$	*bit* $_{16}$	*bit* $_{15}$	80	*overbought*
	1	0	1	0	0	0	0		

Figure 5: An example of transform bits into trading strategy. RSI (14) (20, 80). RSI Look back period starts from 1.

3.2.3 Fitness

The fitness in this paper is the rate of return. When trading strategies are generated, this paper uses simulated transactions to calculate the rate of return of each particle. The higher the rate of return is, the better is the trading strategy.

3.2.4 Quantum Not Gate (QN Gate)

There are three important information we need ot keep track of, which are local best particle, local worst particle and global best particle. After calculating the rate of return in every iteration, the particle with highest rate of return is the local best, the particle with the lowest rate of return is local worst. By sorting the rate of return, it is easy to distinguish which particle is the local best and which is the local worst. We record their rate of return and trading strategy of these two particles. Then we compare the rate of return of local best and global best. If the rate of return the local best is higher than the return rate of the global best, we copy the information of local best to global best, which are strategy and rate of return. Next, according to the information of these global best and local worst, we can now execute the step of quantum not gate. First check each bit of the trading strategy of the global best and local worst. If the global best has $bit_n = 1$, local worst has $bit_n = 0$, and $beta_n$ is below 0.5, then $beta_n = 1 - beta_n$. If the global best has $bit_n = 0$, local worst has $bit_n = 1$, and $beta_n$ is over 0.5, then $beta_n = 1 - beta_n$. As shown in Figure 6. We apply this rule for all 22 bits. This step makes GNQTS algoritm with the ability of escaping the local optimal.

0.65	0.77	0.44	0.5	0.65	0.5	0.51	0.59	*beta matrix*
0	1	1	1	1	0	0	1	*global best*
1	1	0	1	0	0	1	0	*local worst*
QN		*QN*				*QN*		*QN gate*
0.35	0.77	0.56	0.5	0.65	0.5	0.49	0.59	*updated beta matrix*

Figure 6: The process of updating beta matrix with quantum not gate

3.2.5 Update

Delta is used to update the beta matrix when bit_n meets certain conditions. Add delta to $beta_n$ if bit_n of global best is 1 and bit_n of local worst is 0. Subtract delta form $beta_n$ if bit_n of global best is 0 and bit_n of local worst is 1. The example is shown in Figure 7.

	bit_7	bit_6	bit_5	bit_4	bit_3	bit_2	bit_1	bit_0	
···	0.35	0.77	0.56	0.5	0.65	0.5	0.49	0.59	*beta matrix*
···	0	1	1	1	1	0	0	1	*global best*
···	1	1	0	1	0	0	1	0	*local worst*
···	$0.35 - \theta$	0.77	$0.56 + \theta$	0.5	$0.65 + \theta$	0.5	$0.49 - \theta$	$0.59 + \theta$	*updated beta matrix*

Figure 7: The process of updating beta matrix with delta

3.2.6 Termination

When the total iteration has reached or the fitness of global best is satisfied, the process of GNQTS will be terminated; otherwise, the GNQTS will will restart the loop from the measurement step.

3.2.7 Mixing RSI and SMA

There are two technical indicators we use in this research. Both of RSI and SMA are trained separately, in other words, either all the strategies is RSI or SMA. RSI and SMA dose not show up at the same time. To generate more flexable strategies, we proposed a method of combining RSI and SMA together while training.

It is simple to use only one indicator while training. The training result uses the same indicator to form strategies, such as (RSI_{buy}, RSI_{sell}) or (MA_{buy}, MA_{sell}). If mixing two indicators together, the strategies will look like (RSI_{buy}, MA_{sell}) or (MA_{buy}, RSI_{sell}). The proper way of implementing the beta matrix is to split the beta matrix into two part. The total bits of beta matrix will be RSI 22 bits + MA 16 bits = 38 bits. If the strategy we use is (RSI_{buy}, MA_{sell}), the first 22 bits in beta matrix belongs to RSI, the last 16 bits belongs to MA. If the strategy we use is (MA_{buy}, RSI_{sell}), the first 16 bits in beta matrix belongs to MA, the last 22 bits belongs to RSI.

3.3 Sliding Windows

The time span of this research is 10 years, from 2012/01 to 2021/12. If we use only one trading strategy throughout such a long period of time, it will most likely to encounter an overfitting problem. Our trading strategy may not be the most profitable. To avoid this problem, the sliding window method is deployed. The sliding window consist of a training period and a testing period, and it slides according to the testing period up to the end of the investment period. Sliding window can reduce the amount of training data by splitting the training data into several different time frame. Each time frame is trained by GNQTS independently, thus the strategies that GNQTS finds will be better adapted to each period of time. To compare which time frame that GNQTS will find a better rate of return, we use 60 different sliding windows in this research.

In xxx's research, 13 different sliding windows are indroduced, as shown in table 5.

Table 5: 13 kinds of sliding windows

Symmetric	**Asymmetric**	**Year-on-year**
M2M	Q2M H2Q	M*
Q2Q	H2M Y2Q	Q*
H2H	Y2M Y2H	H*
Y2Y		

There are three types of sliding windows, symmetric, asymmetric and year-on-year. The original sliding windows are composed of 4 types of time frame, month (M), quarter (Q), half year (H), and year (Y). The comination of these time frame determine the training period and testing period of a sliding window. As for year-on-year, the training period and testing period are one year apart, because of considering that the business cycle repeated every year for some certain industries. For example, M2M is a sliding window with one month of training period and one month of testing period. Figure 8, 9, and 10 demostrate the symmetric, asymmetric and year-on-year sliding window M2M, H2Q, and H* investing in 2010 and 2011.

<table>
<tr><td colspan="4">M2M</td></tr>
<tr><td>2010</td><td colspan="3">2011</td></tr>
<tr><td>Dec</td><td>Jan</td><td>Feb</td><td>Mar</td></tr>
<tr><td>Train(1)</td><td>Test(1)</td><td></td><td></td></tr>
<tr><td></td><td>Train(2)</td><td>Test(2)</td><td></td></tr>
<tr><td></td><td></td><td>Train(3)</td><td>Test(3)</td></tr>
</table>

Figure 8: Demonstration of M2M sliding window

<table>
<tr><td colspan="12">H2Q</td></tr>
<tr><td colspan="6">2010</td><td colspan="6">2011</td></tr>
<tr><td>Jul</td><td>Aug</td><td>Sep</td><td>Oct</td><td>Nov</td><td>Dec</td><td>Jan</td><td>Feb</td><td>Mar</td><td>Apr</td><td>May</td><td>Jun</td></tr>
<tr><td colspan="6">Train(1)</td><td colspan="3">Test(1)</td><td colspan="3"></td></tr>
<tr><td colspan="3"></td><td colspan="6">Train(2)</td><td colspan="3">Test(2)</td></tr>
</table>

Figure 9: Demonstration of H2Q sliding window

<table>
<tr><td colspan="24">H*</td></tr>
<tr><td colspan="12">2010</td><td colspan="12">2011</td></tr>
<tr><td>Jan</td><td>Feb</td><td>Mar</td><td>Apr</td><td>May</td><td>Jun</td><td>Jul</td><td>Aug</td><td>Sep</td><td>Oct</td><td>Nov</td><td>Dec</td><td>Jan</td><td>Feb</td><td>Mar</td><td>Apr</td><td>May</td><td>Jun</td><td>Jul</td><td>Aug</td><td>Sep</td><td>Oct</td><td>Nov</td><td>Dec</td></tr>
<tr><td colspan="6">Train(1)</td><td colspan="6"></td><td colspan="6">Test(1)</td><td colspan="6"></td></tr>
<tr><td colspan="6"></td><td colspan="6">Train(2)</td><td colspan="6"></td><td colspan="6">Test(2)</td></tr>
</table>

Figure 10: Demonstration of H* sliding window

To better understand the effect of different time period that apply to the sliding window. More time frames are introduced to the training period and testing period. These new times frames are days, weeks, one and a half year, two years, and three years. The newly created sliding windows are as shown in table 6, where D stands for day, W stands for week, and M stands for month. For instance, 20D20 is a sliding window with 20 days of training period and 20 days of testing period.

Table 6: new sliding windows

days		**weeks**	**one and a half year**	**two years**	**three years**
20D20	5D5	4W4	18M18	24M24	36M36
20D15	5D4	4W3	18M12	24M18	36M24
20D10	5D3	4W2	18M6	24M12	36M18
20D5	5D2	4W1	18M3	24M6	36M12
15D15	4D4	3W3	18M1	24M3	36M6
15D10	4D3	3W2		24M1	36M3
15D5	4D2	3W1			36M1
10D10	3D3	2W2			
10D5	3D2	2W1			
	2D2	1W1			

3.4 Normalize Internal Rate of Return (IRR)

The bechmark we use in this research is the internal rate of return. This is a metric used in financial analysis to estimate the profitability of the investment annually. The greater the IRR, the greater the return on an investment.

3.4.1 Training period IRR

There are three types of sliding windows, symmetric, asymmetric, and year-on-year. Considering the overlapping of time frame of asymmetric sliding window, the rate of return (RoR) of training period need to be break down to the smallest unit, which is the daily return rate (DRR), as in formula 4. Then, calculate the average of all the DRR of time frames. At last, the IRR of training period can be computed by the average of all DRR to the power of how many days in a year. The formula for training period IRR is shown in formula 5.

$$DRR = (RoR\ of\ a\ time\ frame)^{\frac{1}{how\ many\ days\ in\ this\ time\ frame}} \tag{4}$$

$$IRR_{training\ period} = (average\ DRR)^{how\ days\ in\ a\ year} \tag{5}$$

3.4.2 Testing period IRR

The IRR of testing period is simple, just calculate the product of each RoR of testing time frame to the power of how many years in the testing period. The formula of testing period IRR is shown in formula 6 , where n is the number of time frames in a sliding window.

$$IRR_{testing\ period} = (\prod_{i=1}^{n} RoRi)^{\frac{1}{how\ many\ years\ in\ the\ testing\ period}} \tag{6}$$

4 Experiment Result

4.1 Experimental Enviroment

4.2 Parameters of GNQTS

Table 7: The parameters of GNQTS

Experimental Parameters	
Initial funds	10,000,000
Test period	2012 to 2021
Particle amount	10
Experiments	50
Iterations	10,000
Delta	0.00016

4.3 Training Period

4.3.1 SMA

4.3.2 RSI

4.3.3 SMA combine RSI

4.3.4 Result of Training Period

4.4 Test Period

4.4.1 SMA

4.4.2 RSI

4.4.3 SMA combine RSI

4.4.4 Result of Test Period

4.5 Self-Analysis

not sure what to write

5 Conclution

Conclution

6 Reference

Referrence