Two Stock-Trading Agents: Market Making and Technical Analysis

Yi Feng, Ronggang Yu, Peter Stone

Department of Computer Sciences
The University of Texas at Austin
http://www.cs.utexas.edu/~{fengyi,ryu,pstone}

Abstract. Evolving information technologies have brought computational power and real-time facilities into the stock market. Automated stock trading draws much interest from both the fields of computer science and of business, since it promises to provide superior ability in a trading market to any individual trader. Trading strategies have been proposed and practiced from the perspectives of Artificial Intelligence, market making, external information feedback, and technical analysis among others. This paper examines two automated stock-trading agents in the context of the Penn-Lehman Automated Trading (PLAT) simulator [1], which is a real-time, real-data market simulator. The first agent devises a *market-making* strategy exploiting market volatility without predicting the exact direction of the stock price movement. The second agent uses technical analysis. It might seem natural to buy when the market is on the rise and sell when it's on the decline, but the second agent does exactly the opposite. As a result, we call it the *reverse* strategy. The strategies used by both agents are adapted for automated trading. Both agents performed well in a PLAT live competition. In this paper, we analyze the performance of these two automated trading strategies. Comparisons between them are also provided.

1 Introduction

With the arrival of the information era, major stock markets such as the NASDAQ are now electronic. The NASDAQ is a distributed trading system completely run through networked computers. It allows customers' best bids and offers to be displayed and represented on the NASDAQ by their brokers or through ECNs (Electronic Crossing Networks), which are electronic trading systems that match buy and sell orders automatically. ECNs such as Island [2], Archipelago [3] and Bloomberg [4] allow customers to display their orders and also allow customer orders to be traded with each other.

ECNs are easy to use, and available to everyone, yet they are not risk-free platforms for customers to test their trading strategies. There are a lot of simulators developed for use without risking money in real markets. The Stock Market Game [5] is a simulator that enables participants to discover the risks and rewards involved in decision-making. Virtual Stock Exchange [6] is another simulator that participants can use to build and manage a virtual portfolio. The

Penn-Lehman Automated Trading (PLAT) simulator [1] uses real-world, real-time stock market data available over modern ECNs. It is the first simulator to incorporate complete order book information, thereby allowing it to "match" agent orders with real-world orders and simulate the resulting effects on the market. It also provides APIs so that the participants can program their strategies and trade with other agents and outside markets automatically.

Using these and other simulators, as well as experiments in the real world, many researchers have studied trading strategies from the perspectives of Artificial Intelligence [7, 8, 12], neural networks [9, 10], technical analysis [11], etc., However, experimenting in the real world is expensive and most simulators differ significantly from real markets such that strategies successful in simulation may not be appropriate for real markets. PLAT is among the most realistic simulators because it both includes real-time, real-world data, and realistically models the effects of the agents' own trades on the market. Thus, we think some interesting, potentially applicable conclusions can be reached based on agent experiments in PLAT.

In this paper, we present two successful automated stock-trading strategies that we have implemented and tested in PLAT. The first agent implements a *market-making* strategy which exploit market volatility without considering the directions of price movement at all. The second one uses technical analysis. It might seem natural to buy when the market is on the rise and sell when it's on the decline, but the second agent does exactly the opposite. As a result, we also call it the *reverse* strategy.[1]

The remainder of this paper is organized as follows. In section 2, the PLAT simulator is described. In section 3 and 4, we describe the agent strategies and analyze their performances. In section 5, we present detailed empirical results, including results from a PLAT live competition and from controlled experiments over a 10-day period. Finally, comparison between these two strategies and further work are discussed in section 6.

2 The PLAT Simulator

The PLAT simulator uses real-world, real-time stock market data for automated trading. It frequently queries the Island ECNs web-site to get the most recent stock prices and buy and sell order books. The simulator then matches the buy orders and sell orders. The orders can be from Island or from trading agents. The simulator also computes the profits and losses of each connected trading agent in real time.

PLAT is equipped for testing strategies on historical data and also for running live competitions. The live competition starts and ends at the same time as normal trading sessions of the NASDAQ. The simulator supports limit orders only. A limit order is a request to buy or sell shares of a stock at a specified price. In the simulation, the best ask price is the lowest price any seller (either

[1] Similar strategies have been termed "contrarian."

trading agents or outside market customers) has declared that they are willing to accept; the best bid price is the highest price any buyer has declared that they are willing to pay. If a new buy order has bid price equal to the best ask price or a new sell order has ask price equal to the best bid price, the order will be matched in the amount of the maximum available shares and the trade is executed. If a bid price is higher than the ask price, the trading price is the average of these two prices. If orders cannot be matched immediately, they are kept in the queue to wait for possible future matches.

Currently, The PLAT simulator is hardwired to Microsoft Stock (Symbol: MSFT). Trading agents in the simulation can buy or sell MSFT stocks with limit orders. There are two types of sales supported in the simulation: long and short. Long sales are what we normally think of when we think of selling, that is, sales of stocks owned by the seller. Short sales are sales of stocks borrowed by the agents from the simulator. Trading agents can also borrow money from the simulator without any interest, however, the same amount of money will be deducted from the agent's simulated cash account. The cash in the account can be negative or positive. The value that a trading agent has in the simulation is calculated in real time by the formula: *value = cash + holdings * current Price.* The cash and holdings are set to 0 at the beginning of the simulation.

The PLAT simulator differs from real markets in several ways:

- There is some time lag in the simulator either due to the server being overloaded or the lag of the data available from the Island ECNs. Resourceful trading agents could potentially derive some advantage from gaining access to a faster real-time data source, but to our knowledge nobody has done so.
- There are no commission or tax charges in the PLAT simulation. Transactions can be executed without any fees. In the real market, too many transactions will increase the overhead cost.
- The trading in PLAT is fully automated, meaning that once strategies are fixed, participants cannot intervene during the trading day.

Our proposed strategies utilize the fact that there are no commission or tax charges in the simulation, and places orders as frequently as possible. In the following sections, we describe two agent strategies and analyze under what market conditions they can make profits.

3 The Market Making Strategy

In this section, we describe our market making strategy and discuss its performance in detail. Several variations of the original strategy are also suggested.

3.1 Basic Approach

The market making strategy we developed exploits the price volatility of a stock, rather than predicting the direction of its movement. It has some similarity with so called "pairtrade", which buys (goes long) a stronger stock and sells (goes short) a weaker stock by research on individual companies. Instead of trading on two different stocks, our market making strategy will place a pair of buy and sell orders of the same volume on a single stock simultaneously, without predicting the stock's future movement. However, we believe that the price of the stock will have a lot of fluctuations, i.e., going up and down frequently during the trading day. The price of the buy order we place will be likely lower than the the current price, while the price of the sell order will be higher. When the price goes beyond our sell order's price, the sell order matches. Then when the price drops back down below the price at which we sold, we will gain profit from the short position. If the price drops down enough and reaches our buy order, we will then buy back with the same number of shares. At that point, we will establish a neutral market position and obtain a profit according to the trade volume and the price gap between our two orders. Similarly, the idea works equally well if the price drops down first and then goes up.

Here is an intuitive example of how this strategy works. Suppose the current price of stock A is \$20.00. We place a buy order of 100 shares at \$19.00 and a sell order of 100 shares at \$21.00. When the price goes beyond \$21.00, our sell order trades and we gain a short position of 100 shares. Obviously, when the price drops back down below \$21.00, our short position will gain profit. If the price drops down enough to reach \$19.00, our buy order trades and now we have a neutral position. The profit we gain is 100 * (\$21.00 - \$19.00) = \$200.00. We place such pairs of orders at every tick of the simulator throughout the trading day. Although we typically use much smaller margins, we expect that with the abundant micro-variation of the stock price, we will be able to accumulate profits over the trading day. Figure 1 demonstrates how our agent works during the simulation.

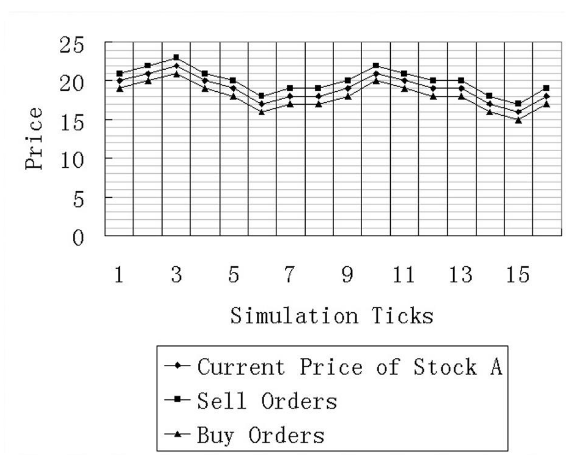

Fig. 1. The Basic Idea of Market Making

A crucial question for this market making strategy is at what price should we place the orders. Note that if we place the order pair far away from the current market price, the orders are unlikely to be executed; on the other hand, if the order pair is too close to the current price, which means they are close to each other, our profit will be compromised. A natural first approach is to use "fixed gaps" related to the current price to set the orders' prices, for example placing the buy order at a price of 0.02 lower than the current price, and the sell order at price of 0.02 higher than the current price. Asymmetric fixed gaps may also be used. But fixed gaps are inflexible and may fail to capture the characteristics of the current market.

After experimenting with a number of fixed gaps, it became apparent that our method is indeed sensitive to the magnitude of the gap. Furthermore, due to the difference in prices and trading patterns for different stocks, it is likely that this value would need to be tuned anew for each different stock. Here we introduce a method for choosing the gaps dynamically based on the current order books. We expect that placing orders with reference to the existent orders in both the buy and sell books will be more robust to a variety of trading scenarios. If and when PLAT expands to include more than one stock we plan to test this hypothesis explicitly. Our strategy currently takes in a parameter n, which varies from 1 to the number of the existent orders. The orders will be placed immediately in front of the nth order in both queues. For example, the buy order we insert will be at \$0.0001 higher than the price of the nth buy order, while the sell order will be at \$0.0001 lower than that of the nth sell order. Table 1 shows a snapshot of the first five orders in both buy and sell order books where the order book oriented market making strategy is used with $n = 2$. The orders in bold outline are the virtual orders placed in the simulator by our strategy, while the others are the real orders in the market.

Buy Order Book		Sell Order Book	
Price	Volume	Price	Volume
24.0360	500	24.0700	350
24.0061	**1000**	**24.0889**	**1000**
24.0060	1500	24.0890	600
24.0010	800	24.0950	2000
23.9700	1000	24.0950	1200

Table 1. Buy and Sell Order Books

Table 2 gives the pseudo-code of this basic approach. Note that the volume in the order is also an important parameter.

In principle, n is a good candidate for automatic tuning using machine learning techniques. We leave that for future work. Here we use a value of $n = 1$ for the competition and controlled experiments discussed in later sections.

while time permits:
buyReferencePrice $\leftarrow$ getBuyOrderPrice(n) + 0.0001;
placeOrder(BUY, buyReferencePrice, volume);
sellReferencePrice $\leftarrow$ getSellOrderPrice(n) - 0.0001;
placeOrder(SELL, sellReferencePrice, volume);

Table 2. The Order Book Oriented Market Making Strategy

3.2 Performance Analysis

When using the market making strategy, we expect that our agent will accumulate profit when the stock price has a lot of fluctuation over the trading day. If all trades match, our profits are the average differences between prices of the order pairs times the number of the simulation ticks. The ideal situation for our strategy is when the end price of a day is very close to the start price. That means we can have our share position close to neutral, and more pairs of our orders executed. On the contrary, if the stock price moves drastically in one direction (either high or low) and doesn't come back before the end of the day, our strategy will lose money. This becomes clear after some microanalysis. For the simple example we discussed above, when the price drops below \$19.00, our buy order will trade and we will have a long position of 100 shares. If the price keeps dropping and never comes back, the value of the 100 shares will be less than \$1,900.00 and we will be losing money. Intuitively, such situation will propagate in our strategy, when the price keeps dropping and we keep placing orders. Figure 2 and Figure 3 are the price charts of MSFT on March 21st and March 24th, 2003, which are examples of a "good" day and a "bad" day towards the strategy respectively.

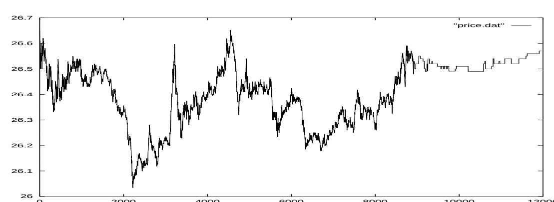

Fig. 2. March 21st 2003, A "Good" Day

Based on the market making strategy we are using, our agent tends to get into a large short position if the price goes up consistently; on the other hand, a large long position if the price goes down stiffly. Either of these situation will lead to a loss in value, according to the discussion above. Furthermore, the value will be at the mercy of the huge position, and the profit accumulated by our strategy will be dominated by it. That is, for instance, if we hold a lot of volume and the stock price goes down, our resulting losses dominate the small profits we

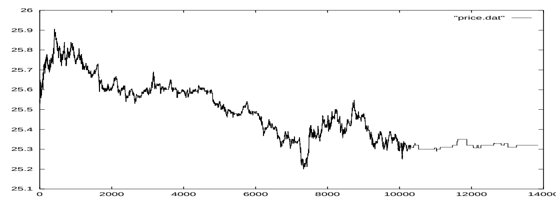

Fig. 3. March 24th 2003, A "Bad" Day

get from each matched pair of trades. To mitigate such a problem, we can try to prevent our holdings from going to one extreme or the other. We have designed two methods, which are based on the two most important characters of an order - price and volume, to achieve a better control on our share position.

Our original approach sets the pair of orders at "semi-symmetric" price, placing orders right in front of the nth order in both queues. In order to keep our share position from going to one extreme, we can vary the price of the orders to encourage the opposite trades. For instance, if we now have a large positive holding, we may lower the price of the sell orders calculated by the original method, to encourage sell trades to happen, and vice versa. We name this the *price control method*. A new positive floating point parameter, priceEncouragement, is introduced to accomplish this task. It will encourage the opposite trade according to the current position in a continuous way. The pseudo-code of the price control method is given in Table 3. The currentPosition, which can be a positive or negative integer, records the current holdings of the agent.

```
while time permits:
buyReferencePrice ← getBuyOrderPrice(n) + 0.0001;
sellReferencePrice ← getSellOrderPrice(n) - 0.0001;
currentPosition ← getAgentCurrentPosition()
if currentPosition < 0
    buyReferencePrice ← buyReferencePrice - currentPosition * priceEncouragement
elseif currentPosition > 0
    sellReferencePrice ← sellReferencePrice - currentPosition * priceEncouragement
placeOrder(BUY, buyReferencePrice, volume);
placeOrder(SELL, sellReferencePrice, volume);
```

Table 3. The Price Control Method

Trade volume is the other perspective from which we can manage our position. In the original method, we have equal trade volumes in every pair of orders we put into the market. However, we can alter the trade volume in both orders with regard to the agent's current share position. For example, if we have positive holdings, we may increase the trade volume of the sell order and decrease that of the buy order for the next pair we insert, and vise versa. This is the *volume control method* we propose to alleviate the huge position problem.

We introduce a positive floating point parameter, volumeAlteration, to tune the trade volume of both orders in a continuous way. Table 4 shows the pseudo-code for the volume control method.

```
while time permits:
buyReferencePrice ← getBuyOrderPrice(n) + 0.0001;
sellReferencePrice ← getSellOrderPrice(n) - 0.0001;
currentPosition ← getAgentCurrentPosition()
buyVolume ← volume * (1 - currentPosition * volumeAlteration)
sellVolume ← volume * (1 + currentPosition * volumeAlteration)
placeOrder(BUY, buyReferencePrice, buyVolume);
placeOrder(SELL, sellReferencePrice, sellVolume);
```

Table 4. The Volume Control Method

We also combine the price control and the volume control methods into a new strategy, which will offer more combinations of parameters in the experiments. However, all these control methods may compromise the profit accumulated through the original strategy. This is because they either decrease the price gap between the pair of orders, or decrease the trade volume, or both. They are expected to reduce losses on some "bad days", and can be considered more conservative options of the original market making strategy. But, as demonstrated by numerous experiments, there are also exceptions where the control methods obtain larger profit or lead to bigger losses.

4 The Reverse Strategy

In this section, we first describe the basic and reverse strategies. The reverse strategy is used by the second agent. Then we analyze under what conditions the reverse strategy will make profits using a realistic but simplified price model.

4.1 Strategy Description

Our initial *basic* strategy for the second agent is as follows. At any time during the simulation, if the stock price goes up, it places a buy order; and if the stock price goes down, it places a sell order. The motivation is that a price rise indicates likely further price rises. However, initial testing of this strategy revealed that it lost money more often than it gained. As a result, we decided to flip the buy and sell order conditions. We call the resulting strategy the *reverse* strategy since it does exactly the opposite of the initial strategy. Table 5 and Table 6 show the pseudo-codes for the basic and reverse strategies.

The reason that the reverse strategy makes profits in many kinds of market is that stock market prices are not constant and in fact undergo frequent changes in direction, rather than moving consistently in one direction. On most trading

```
while time permits:
lastPrice ← getLastPrice();
currentPrice ← getCurrentPrice();
if currentPrice > lastPrice
    placeOrder(BUY, currentPrice, volume);
elseif currentPrice < lastPrice
    placeOrder(SELL, currentPrice, volume);
```

Table 5. The Basic Strategy

```
while time permits:
lastPrice ← getLastPrice();
currentPrice ← getCurrentPrice();
if currentPrice > lastPrice
    placeOrder(SELL, currentPrice, volume);
elseif currentPrice < lastPrice
    placeOrder(BUY, currentPrice, volume);
```

Table 6. The Reverse Strategy

days, there are a lot of small spikes in the stock price chart. Figure 4 shows such an example. Under these conditions, we expect to make profits with the reverse strategy, as described in the next subsection.

Fig. 4. MSFT Stock Price: Dec. 20, 2002

4.2 Performance Analysis

For any fixed strategy (other than the degenerate do-nothing strategy), there are some market conditions in which it will make money and some under which it will lose money. We analyze the performance of the basic and reverse strategy based on what we think are realistic, though simplified, assumptions about the price trajectories. We assume that the price of MSFT oscillates consistently in a wave pattern around a constant price p. We call the microvariation of the price (the wave's amplitude) during the trading day δ. Figure 5(a) illustrates such a price trend. We also assume that all orders placed in buy queues and sell queues will be matched at some point during the trading day (perhaps immediately).

In our analysis, we assume that the start price is $p - \delta$. The start price is not crucial to the analysis since if it's not $p - \delta$, we can ignore the first several ticks until we have the price $p - \delta$. Tables 7 and 8 show the calculations of the changes in holdings, cash and value for the basic and reverse strategy in one cycle. Each time we place a sell order or buy order, the volume size is v.

time	price	action	holding	cash	value
t_0	$p - \delta$	—	0	0	0
t_1	p	buy	v	$-vp$	0
t_2	$p + \delta$	buy	$2v$	$-2vp - v\delta$	$v\delta$
t_3	p	sell	v	$-vp - v\delta$	$-v\delta$
t_4	$p - \delta$	sell	0	$-2v\delta$	$-2v\delta$

Table 7. The Basic Strategy over One Cycle.

time	price	action	holding	cash	value
t_0	$p - \delta$	—	0	0	0
t_1	p	sell	$-v$	vp	0
t_2	$p + \delta$	sell	$-2v$	$2vp + v\delta$	$-v\delta$
t_3	p	buy	$-v$	$vp + v\delta$	$v\delta$
t_4	$p - \delta$	buy	0	$2v\delta$	$2v\delta$

Table 8. The Reverse Strategy over One Cycle

From Tables 7 and 8, we can see that the basic strategy loses \$$2v\delta$ in one cycle. After k cycles, it loses \$$2vk\delta$. On the other hand, the reverse strategy earns \$$2vk\delta$: the value increases linearly over time.

Although the preceding calculation relies on a very restrictive model of market dynamics, it is a pattern that seems to be reasonably representative of reality. Next, we will consider a relaxation to our initial price trajectory assumption: the price of MSFT oscillates around a constant price p.

As is apparent from Figure 4, it is much more common that the price oscillates around a line with a non-zero slope than that it oscillates around a constant price. We define the price trend as a line: $y = ax + b$. a is the slope, while x is the time elapsed from the start of the simulation, which is measured by the number of simulation ticks. If the tick number is odd, price=y. If it is divisible by 4, price $= y - \delta$. If it is even, but not divisible by 4, price $= y + \delta$. Figure 5(b) illustrates such a price trend. Table 9 shows the calculation of the reverse strategy. In the simulation, δ is always positive, while a can be positive or negative. Since the basic strategy does the exact opposite of the reverse strategy, we omit the calculation.

There is one additional assumption in Table 9's calculation: $\delta > |a|$, guaranteeing that the price actually oscillates around the line, rather than moving

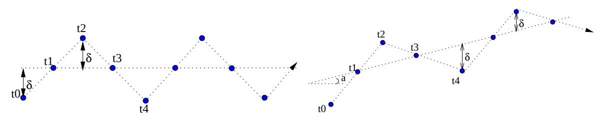

Fig. 5. Different price trend models: (a) shows the initial model in which the price oscillates consistently in a wave pattern around a constant price. (b) extends the model to allow for the price to oscillate around a line with non-zero slope.

time	price	action	holding	cash	value
t_0	$b-\delta$	—	0	0	0
t_1	$a+b$	sell	$-v$	$v(a+b)$	0
t_2	$2a+b+\delta$	sell	$-2v$	$v(3a+2b+\delta)$	$v(-\delta+a)$
t_3	$3a+b$	buy	$-v$	$v(b+\delta)$	$v(\delta-3a)$
t_4	$4a+b-\delta$	buy	0	$2v(\delta-2a)$	$2v(\delta-2a)$

Table 9. The Calculation of the Reverse Strategy: Linear Model

consistently in one direction. After k cycles, the value of the reverse strategy is $\$2vk(\delta-2a)$. Combining this formula with the preceding assumption, we see that if $-\delta < a < \delta/2$, the reverse strategy makes a profit in the linear model; otherwise it loses money. Note that If $a = 0$, the result degenerates to that of the constant price model, as expected. Note also that this analysis can be easily extended to the case where the underlying price trajectory is *piece-wise linear*: each line segment can be considered independently.

In general, if the price trend is not biased, i.e. the price goes up and down consistently, the reverse strategy will make profits, otherwise, it will lose money.

5 PLAT Live Competition and Experimental Results

In this section, we present detailed empirical results demonstrating the behavior of our agent strategies under real market conditions. First, we describe their performances in a PLAT live competition that included several agents in addition to our two agents. Then we present several experiments run on historical data with only our own agents participating separately.

The PLAT live competition was held over 15 trading days from February 24th to March 18th 2003. It averaged the performance over many simulations to get more reliable indications of the performances of the trading strategies. The competition was structured as a 3-week, 3-round tournament with one week (5 trading days) for each round. The participants were 25 students with strong interests in the project. In the first round, 13 teams were formed and divided into two pools: the red pool and the blue pool. The total returns for the first round were aggregated and the three top scoring teams from each pool advanced to the second round. In the second round, the 6 remaining teams were run in a single simulation. The top two performing teams in the second round met in the finals. In order to limit the divergence of the simulation from the real market, each agent was restricted to maintain holdings within 100,000 of 0 (positive or negative). Violations were grounds for disqualification from the competition.

The market making strategy used by the first agent included the price control mechanism with priceEncouragement=0.01. For the second agent using the reverse strategy, we set our trade volume to 1000 shares per order in the competition. This volume size was chosen without detailed experimentation on the one hand to be as large as possible, but on the other hand in the hopes that it was small enough that orders would be matched fully. We also introduced a simplified holding control mechanism to avoid violating the 100,000 share limit. When the holdings exceed 85,000 or are lower than -85,000, the reverse strategy stops placing buy orders or sell orders respectively until the holdings are back to a normal level. If the absolute value of holdings exceeds 95,000, the reverse strategy cancels all the orders placed in the buy queue or the sell queue.

In the first round of the PLAT live competition, the market- making strategy was placed in the red pool, and was Team 8. The reverse strategy was placed in the blue pool and was Team 5. Tables 10 and 11 show the results from the first round. Team 11 in the red pool was disqualified after three trading days because of repeated and extreme violation of the share position limit. Team 7 had third-place earnings in the red pool, but was disqualified for its share position limit violations three times in five days. The red pool qualifiers were Teams 12, 8 (our market-making strategy) and 13. The blue pool qualifiers were Teams 4, 5 (our reverse strategy) and 1.

Team	24-Feb	26-Feb	27-Feb	28-Feb	3-Mar	total	Rank
Team 1	-479	3711	17603	1278	-266	21847	3
Team 2	334	2841	-53	-464	338	2996	5
Team 3	1046	3845	1980	379	1644	8894	4
Team 4	100557	-3104	-7314	-1642	-4360	84137	1
Team 5	**12489**	**26357**	**-4304**	**12236**	**-346**	**46432**	**2**
Team 6	-15228	-79442	-2052	4218	-20816	-113370	6

Table 10. PLAT Live-data Competition Round 1, Blue Pool. (Team 5: our reverse strategy)

The six qualifiers advanced to the second round, facing off in a single pool over a five-day trading period to determine the two finalists to meet in the third round. The second round ran from March 5th to March 11th 2003. Table 12 shows the results of the second round.

The two top scoring teams (Teams 12 and 5) from the second round advanced to the finals. They were the only two teams in the black after the second round competition. Table 13 shows the result of the third round competition.

Despite big losses in the third round due in part to some uncommon price patterns, our reverse strategy lost less money than our competitor, and won the competition. The fact that the relative rankings of team 5 and 12 change could

Team	24-Feb	26-Feb	27-Feb	28-Feb	3-Mar	total	Rank
Team 7	-23935	20338	20791	50949	-28460	39683	3
Team 8	**-9590**	**8200**	**79008**	**21357**	**-20795**	**78180**	**2**
Team 9	2243	4414	6038	2103	3072	17870	6
Team 10	532	-2915	38396	-549	-3088	32376	5
Team 11	-4141632	-3434560	-31561216	N/A	N/A	N/A	7
Team 12	20187	29090	54001	7208	16673	127159	1
Team 13	-15066	8572	42734	25170	-26168	35242	4

Table 11. PLAT Live-data Competition Round 1, Red Pool. (Team 8: our market-making strategy)

Team	5-Mar	6-Mar	7-Mar	10-Mar	11-Mar	total	Rank
Team 8	**22433**	**18640**	**-35475**	**-9826**	**-4156**	**-8384**	**4**
Team 12	-14840	27982	-28438	13257	23120	21081	1
Team 13	-4223	17731	-40912	-18271	21933	-23742	5
Team 1	-563	-395	-967	-776	-486	-3187	3
Team 4	-6700	6	-30608	83	-135	-37354	6
Team 5	**11095**	**12105**	**-21931**	**10399**	**-9979**	**1689**	**2**

Table 12. PLAT Live-data Competition Round 2. (Team 5: our reverse strategy; Team 8: our market-making strategy)

Team	12-Mar	13-Mar	14-Mar	17-Mar	18-Mar	total	Rank
Team 12	-54156	-79233	-21896	-18032	-886	-174203	2
Team 5	**-33382**	**-54611**	**6915**	**-71397**	**26304**	**-126171**	**1**

Table 13. PLAT Live-data Competition Round 3. (Team 5: our reverse strategy)

be a result of several different factors. For one thing, the economy is different without the other agents participating. In addition, the real market differed during the two runs. In particular, the market was very bullish during the finals, a condition that, ironically, hurt both agents, but agent 12 slightly more.

Figures 6 and 7 show the performance of the reverse strategy on two trading days during the finals. First, on Mar. 17, 2003, Wall Street had its best day since Oct. 2002. However, it was a disaster for our reverse strategy. The MSFT stock price increased consistently on that day, which, as per our analysis, is the worst case scenario for the reverse strategy. As a result, we lost a lot of (simulated!) money. In contrast, on Mar. 18, 2003, The MSFT price roughly oscillated around a constant line, and our reverse strategy finished in the black.

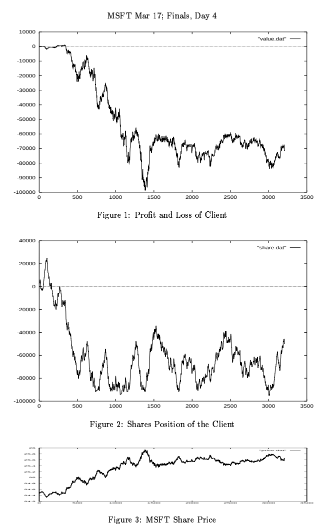

Fig. 6. Reverse Strategy in PLAT Live-data Competition Round 3, Mar. 17, 2003

The market making strategy also did well at the beginning of the Round 2 in the competition. This strategy too favors a trade day with lots of price fluctuation and relatively small difference between starting and ending price. Figures 8 and 9 show the performance of the maket makeing strategy on March 6th and 7th during Round 2.

Besides the live-data competition, we also did some historical experiments with our agent strategies. Table 14 shows the results. Column 2 shows the value of the market-making strategy and Column 3 shows the value of the reverse strategy. Both strategies ran on the historical dates separately.

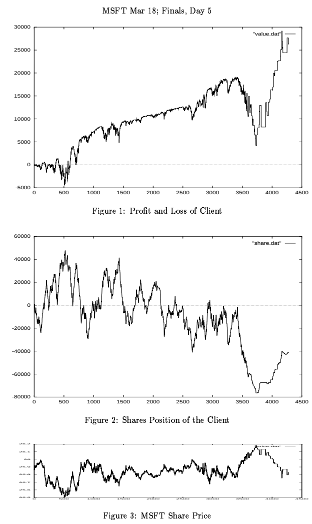

Fig. 7. Reverse Strategy in PLAT Live-data Competition Round 3, Mar. 18, 2003

DATE	Market Making	Reverse
Mar21, 2003	25557	7730
Mar24, 2003	-30845	-21602
Mar25, 2003	-4742	-4504
Mar26, 2003	28453	-5525
Mar27, 2003	5856	-21932
Mar28, 2003	13174	-7146
Mar31, 2003	8520	-3489
Apr01, 2003	31144	6500
Apr02, 2003	-34444	6087
Apr03, 2003	-31938	30929

Table 14. Historical Experiments

From Table 14, we can see that both strategies lost money some days and made profits on other days. We did our experiments on dates from Mar 21st to April 3rd 2003. We chose these dates before seeing the results and report all of the results here. The MSFT stock price endured high variability during these days due to the affairs related to the war in Iraq. We expect the performances of our agent strategies to be much better if the MSFT stock price is more stable and consistent. Meanwhile, risk control mechanisms should be emphasized to make the strategies more adaptable and profitable.

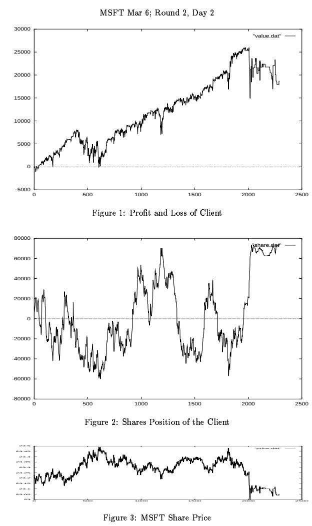

Fig. 8. Market Making Strategy in PLAT Live-data Competition Round 2, Mar. 06, 2003

The price control, volume control and combined control methods for the market making strategy were also tested during the same period. There are small differences between the implementations of the control parameters and those described in the pseudo-codes in Section 3, since we need to cope with the position limit imposed by the PLAT live competition. In the price control method and the combined control method, the price of the buy orders is calculated as follows:

buyReferencePrice ← buyReferencePrice -
currentPosition / positionLimit * priceEncouragement

And the trade volume of the buy orders in the volume control method and the combined control method is calculated as follows:

buyVolume ← volume * (1 - currentPosition / positionLimit * volumeAlteration)

Similar changes are applied to the sell orders in each control method. For all the tests, we used 1000 shares as the volume parameter. 0.01 was used as the priceEncouragement parameter, which implies the largest possible price encouragement will be 0.01 when the share position is at the limit. We picked 0.5 as the volumeAlteration parameter, which implies the largest possible volume alteration will be 500 shares when the share position is at the limit, according to the 1000 shares volume. These parameters were all chosen based on our own informal observations during preliminary testing. They were not tuned extensively. The results are shown in Table 15 with the original strategy.

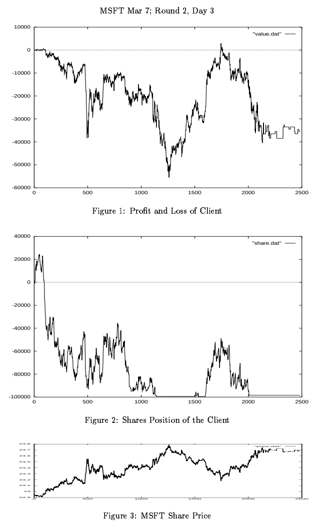

Fig. 9. Market Making Strategy in PLAT Live-data Competition Round 2, Mar. 07, 2003

DATE	Original MM	Price Control	Volume Control	Combined Control
Mar21, 2003	25557	41553	40085	41030
Mar24, 2003	-30845	-23751	-34957	-23139
Mar25, 2003	-4742	-20407	-15353	-20884
Mar26, 2003	28453	12191	28402	9573
Mar27, 2003	5856	-16540	5489	-6504
Mar28, 2003	13174	-1036	1526	-2134
Mar31, 2003	8520	7148	8933	15085
Apr01, 2003	31144	24121	27028	26620
Apr02, 2003	-34444	-15797	-31844	-25801
Apr03, 2003	-31938	-23900	-36376	-34720

Table 15. Experiments on Variations of Market Making Strategy

From Table 15, we can see that normally all the variations of market making strategies behave similarly – either all ending in red or all ending in black. However, exceptions such as Mar 27th and 28th do exist. Overall, no strategy dominates all the others. They offer a good variety when we trade with market making strategies.

6 Discussion and Future Work

The market making strategy and the reverse strategy use different approaches in the PLAT simulator, yet they both performed well in the PLAT live competition. They both utilize the fact there are a lot of small spikes in the MSFT price chart, and make profits little by little when placing each order or pair of orders. The market making strategy is a complicated strategy using order book information which is the unique feature of the PLAT simulator.

In the PLAT simulation, there are no tax or commission charges. We utilize this fact and place buy orders and sell orders as frequently as possible. For real markets, there may exist commission or tax charges each time a transaction is executed. Since these charges are entirely predictable in our models, it is a simple matter to compare expected profits of these two strategies against expected charges. When the latter exceeds the former, then these two strategies should not be used.

We have limited information about the strategies used by other agents in the PLAT competition. but some keywords used to describe them on the website [13] are "crossover moving average," "intersecting the geometric trend," "case-based learning," and "Static Order Book Imbalance."

From the historical experiments, we can see that both strategies are less adaptable to external factors like important live news release. We hope that we can use the external data indication as risk control mechanism for both of these two strategies in the future.

7 Acknowledgments

We thank Professor Michael Kearns and his PLAT group at the University of Pennsylvania for their development of the PLAT simulator and for allowing us to participate in the competitions. This research was supported in part by NSF CAREER award IIS-0237699.

References

1. The Penn-Lehman Automated Trading Project. Michael Kearns and Luis Ortiz. *IEEE Intelligent Systems* 18(6):22–31. November/December 2003.
2. Island: http://www.island.com
3. Archipelago: http://www.tradearca.com
4. Bloomberg: http://www.bloomberg.com
5. The 2002 annual report of Securities Industry Foundation for Economic Education, 2001: http://www.sia.com/about_sia/pdf/annual2002.pdf
6. Virtual Stock Exchange: http://www.virtualstockexchange.com
7. R. Freedman et. al., eds., *Artificial Intelligence in Capital Markets*, Chicago, IL: Probus Pub., 1995.
8. A. Skabar and I. Cloete. "Discovery of Financial Trading Rules." In *Proc. Artificial Intelligence and Applications*, pp. 121-125, 2001.

9. J. Kingdon. *Intelligent Systems and Financial Forecasting.* New York, NY: Springer, 1997.
10. R. Donaldson and M. Kamstra, "Forcast Combining With Neural Networks", *Journal of Forecasting*, vol. 15, No. 1, pp.49-61. 1996.
11. M. Sheimo. *Stock Market Rules: 50 of the most widely held investment axioms explained, examined and exposed*, Chicago, IL: Probus Pub., 1991.
12. P. Stone and A. Greenwald, "The First International Trading Agent Competition: Autonomous Bidding Agents". Electronic Commerce Research 5(1), 2005 (To appear).
13. The Penn-Lehman Automated Trading Project: http://www.cis.upenn.edu/~mkearns/projects/pat.html