Automated Stock Trading in PLAT

Alexander Sherstov

December 4, 2003

Abstract

This report documents the development of an autonomous stock trading agent within the framework of the Penn-Lehman Automated Trading (PLAT) simulator. The three approaches presented take inspiration from reinforcement learning, myopic trading using regression-based price prediction, and market making. The performance of these approaches is assessed separately using a fixed opponent strategy, SOBI. Controlled experiments are described that isolate the effects of individual parameters of each strategy. Finally, a comparative analysis of the strategies is presented and suggestions are discussed for future work.

1 Introduction

Automated stock trading is a burgeoning research area with important practical applications. The advent of the Internet has radically transformed the mechanics of stock trading in most stock exchanges. Traders can now readily purchase and sell stock from a remote site, using Internet-based order submission protocols. Even more importantly, traders can monitor the contents of buy and sell books in real time using a Web-based interface. The electronic nature of the transactions and the availability of up-to-date buy and sell book data make autonomous stock-trading applications a promising alternative to immediate human involvement. The Penn-Lehman Automated Trading (PLAT) project at the University of Pennsylvania is an example of a research initiative designed to provide a realistic experimental testbed for stock-trading strategies. PLAT provides a simulated stock trading environment that merges virtual orders submitted by computer programs with real-time orders from the Island stock exchange. No actual monetary transactions are conducted, and the efficacy of a trading strategy can be reliably assessed in the safety of a simulated market.

The problem addressed in this project is one of trading stock throughout a trading day so as to maximize the aggregate profit at the end of the day, with the additional requirement that the share position be completely unwound (i.e., any owned shares liquidated and any owed shares bought back). This work is based on the assumption that profit maximization and position unwinding are two distinct objectives that the automated trading application can treat separately. Specifically, the discussed approaches are designed to maximize profit at the end of the simulation period, under the assumption that the position can be unwound at no great cost shortly before the close of the market. A complete trading application would run one of the trading strategies described below from 9:30 am, the normal opening time, to, say, 3:00 pm, an hour before the market close. In the concluding hour of the trading day, control would be turned over to a simple position unwinding module whose objective would be to continually match owned or owed shares with the top orders in the corresponding order book. Although the delegation of profit maximization and position unwinding to distinct modules may be a suboptimal task decomposition, it greatly simplifies automated trader design. Moreover, concrete profit-maximization strategies proposed in this paper, with perhaps the exception of the RL-based approach, hold very reasonable share positions throughout the day, making unwinding feasible at a tolerable

cost.

The ensuing sections present three approaches to automated stock trading using PLAT and offer experimental results that provide a measure of their performance both in isolation against a fixed opponent and in a joint simulation. The remainder of the paper is organized as follows. Sections 2, 3, and 4 focus on the development and evaluation of the individual strategies, and Section 5 provides a comparative experimental study of their performance in the same market. In all cases strategy performance is evaluated on a set of 10 trading days, carefully selected to be representative of typical price dynamics. The price graphs corresponding to each of the 10 days can be found in Appendix A. Finally, Section 6 concludes with a discussion of unresolved questions and promising directions for future work. For brevity, the term "agent" is used throughout the paper to refer to an autonomous trading application.

2 Reinforcement Learning: A Model-free Approach

This section presents a trading strategy based on reinforcement learning (RL). A brief overview of reinforcement learning is provided below, followed by a section detailing the design of the strategy and an evaluative analysis. Strengths and weaknesses of this methodology are assessed in light of the experimental results.

2.1 RL Overview

RL is a model-free machine-learning technique for achieving good long-term performance in poorly understood and possibly non-stationary environments. Given the seemingly random market trends and fluctuations, it is tempting to resort to a model-free technique designed to optimize performance given minimal domain expertise and a reasonable measure of progress (the reward function).

In its simplest form, a reinforcement learning problem is given by the 4-tuple $\{S, A, T, R\}$, where S is a finite set of distinguishable states of the environment; A is a finite set of actions available to the agent as a means of extracting an economic benefit from the environment, referred to as a *reward*, and of possibly altering the environment state; $T : S \times A \to S$ is a state transition function; and $R : S \times A \to \mathrm{R}$ is a reward function. The objective is to develop a *policy*, i.e., a mapping from environment states to actions, that would maximize the long-term return obtained by the agent. A common definition of return, and one used in this work, is the discounted sum of all rewards obtained by the agent throughout its interaction with the environment: $\sum_{t=0}^{\infty} \gamma^t r_t$, where $0 < \gamma < 1$ is a discount factor.

The original RL framework was designed with a discrete state-action space in mind. In order to accommodate the continuous nature of the problem, the approach described here uses a linear *function approximation* method, tile coding, to allow generalization to unseen instances of the continuous state-action space. For further details on reinforcement learning, the interested reader is referred to [2].

2.2 Strategy Design

The stock trading problem is to buy and sell shares throughout the day so as to maximize the aggregate profit at the end of the trading day, with the additional requirement that the share position be completely liquidated by the time the market closes. I have explored several formulations of the stock trading problem as a reinforcement-learning task. Ensuing is a list of parameters that are potentially helpful as state variables. Agent-specific parameters are typeset in lowercase; global parameters and market statistics are typeset in uppercase.

- *Share holdings*, s. This is arguably the most important component of the agent's state. For example, a hypothetical optimum trading strategy would avoid accumulating excessive share holdings when the price is consistently declining. Additionally, knowledge of the share position is important because of the liquidation requirement.

- *Unmatched volume in the books*, v_{sell} and v_{buy}. This information can be helpful, e.g., for deciding when to withdraw unmatched shares if the

market conditions change so as to make the execution of the submitted orders undesirable.

- *Simulator/Island last price*, P_{sim} and P_{Isl}. Knowledge of the last price is necessary for appropriate order pricing and for detecting market trends (rise/decline).

- *Liquidity measure*, $L = \frac{dV}{dT}$, where V is the total matched volume. This parameter is a measure of how liquid the market currently is. In particular, it may be undesirable to hold large positive or negative share positions in an illiquid market: the impossibility of timely liquidation may have serious consequences due to the unwinding requirement as well as the highly stochastic nature of the stock market itself.

- *Buy and sell quartiles*, Q_{sell} and Q_{buy}. These market statistics can be viewed as an "expression of market sentiment" and as such "may provide strategic guidance for order placement" ([1], section 2.1).

Somewhat counter-intuitively, the cash position is of no use as a state variable: the "right" trading decision is never contingent on the cash holdings because the agents are allowed to have an arbitrarily large negative cash balance. Likewise, a "remaining time" parameter would not be helpful because position unwinding is not part of the strategy, as discussed in the introduction.

To keep the learning task manageable, I identified the following small collection variables as a sufficiently detailed description of the state:

- $\Delta P = P_{sim} - \overline{p}$, the price difference parameter computed as the difference between the current last price and an exponential average of previous last prices: $\overline{p}_i = \beta \overline{p}_{i-1} + (1 - \beta) P_{sim}$. The effect of β is to make the agent focused on short-term trends or long-term trends (see Section 2.3 for an experimental study of this effect). The definition of the price parameter as a difference serves a twofold purpose. On the one hand, it gives an indication of the latest market trend: $\Delta P \approx 0$ corresponds to a stationary market (the price is nearly the same, with balanced random variations in either direction), $\Delta P < 0$ corresponds to a decline in price, and finally $\Delta P > 0$ reveals that the stock price is on the rise. On the other hand, this definition of the price parameter makes the learned policy more general by eliminating the dependency on the absolute value of the stock. The absolute value of the last price would make a meaningless state variable ($25.89 may be an unusually low price for a day when the stock is selling for $27.50 and at the same time an unusually high price on a different day).

- s, the agent's current share holdings. The relevance of this parameter has been discussed at length above.

Little is lost by limiting the state to the above two variables. For example, unmatched volume is of no vital importance: even if the market conditions change unfavorably and the unmatched volume is matched later on, this will be reflected in the agent's share position s, and the agent will take remedial action. By excluding this parameter, two dimensions are eliminated (v_{sell} and v_{buy})—a significant savings. Likewise, a single price parameter is enough to capture the price dynamics in both the real and virtual markets since the virtual market closely follows Island ([1], section 4.2.4). Furthermore, the buy and sell quartiles are of no great predictive value because, once again, the virtual market follows very closely the Island dynamics, over which it exercises no control; therefore, concentrating on the price alone can be viewed as an acceptable omission. Lastly, the liquidity measure is also arguably superfluous: it the market is liquid, the agent's orders will readily execute and will affect its share position accordingly; if the market is illiquid, the share position (and hence the agent's value) will remain unaffected. The only complication arises when the agent purchases too much stock in an illiquid market and cannot unwind its position before the end of the day. In my view, this failure to account for the liquidity is more than compensated for by the corresponding decrease in complexity.

Given the state definition above, the action space is defined as follows:

Volume v of shares to purchase ($v > 0$) or sell ($v < 0$).

It would seem natural to augment the action space to include a price at which to buy/sell, but this extension can be foregone at no significant cost: if the submitted price is always set to the last price, the agent should be able to adjust the demanded volume accordingly. At the same time, this omission allows a dimension of complexity to be eliminated.

With the state and space actions defined, all that is left is to formulate a reward function. Ideally, the reward should be computed only once, at the end of the trading day, with zero intermediate rewards assigned at each time step along the way; otherwise, there is a danger that the trader will learn to optimize the sum of local rewards without optimizing the final position ([2], chapter 3). There are two important objections to this approach. First of all, this formulation of the reward function makes on-line learning impossible: the agent must wait until the end of the trading day to evaluate its actions. This is a nontrivial limitation because the agent may be called upon to function in very diverse economies, some of which may not be well captured by the training environment. Real-time adjustment to an altered economy would be a very desirable capability. The second objection is of a practical nature. Given the complexity of the state-action space (3 continuous variables) and the duration of a simulation (≈50000 time steps in live mode), the training time requirements of this method seem excessive, even if eligibility traces are used to accelerate convergence. A reasonable definition of local reward (namely, difference in present value, computed as cash holdings plus the value of the shares at the Island last price) promises to yield a policy of comparable quality with much less training effort. The ensuing sections explore several local reward functions and evaluate their effectiveness.

2.3 Parameter Choices

To make comparative analysis more meaningful, I used the same parameters in all strategies presented below. Specifically, I used a learning rate of $\alpha = 0.04$, a discount rate of $\gamma = 0.8$, an exploration rate of $\epsilon = 0.1$, and eligibility traces with $\lambda = 0.7$. I have not experimented with varying these values and used them as reasonable general settings. The exploration rate and the learning rate were kept constant at all times because decaying them would lead to convergence, which is undesirable given the non-stationary nature of the environment.

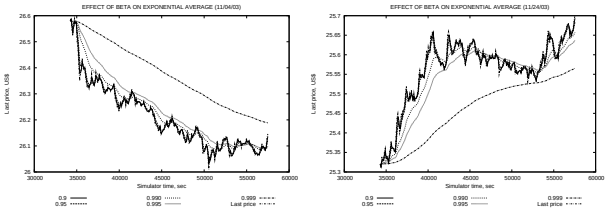


Figure 1: Comparison of $\beta = 0.9, \beta = 0.95, \beta = 0.990, \beta = 0.995, \beta = 0.999$

A final parameter that played a central role was β, the update rate for computing the exponential average of past prices: $\overline{p}_i = \beta \overline{p}_{i-1} + (1 - \beta) P_{sim}$. Figure 1 demonstrates the behavior of the average price on two trading days with different price dynamics and β settings. The values compared are 0.9, 0.95, 0.990, 0.995, and 0.999. As the graphs indicate, the closer β is to 1, the more "inert" the exponential average, i.e., the less responsive to changes in the price trend. On the one extreme, $\beta = 0.9$ essentially duplicates the last price graph, yielding little information about past price dynamics. On the other extreme, $\beta = 0.999$ yields an average that is not representative at all of the changes in price dynamics. The graphs indicate that a choice of $\beta = 0.99$ offers a nice balance, responding sufficiently quickly to genuine trend reversals and ignoring random fluctuations. There is a nuance, however, regarding the use of β under different order placement frequencies. In historical mode, the agent can place much fewer orders per time unit than in live mode. In the graphs of Figure 1, β was tuned based on a frequency of 1 order per 10 seconds. To adjust for the discrepancy, the actual value of the exponential-average parameter is computed as $\beta^{(t_{current} - t_{previous})/10}$, where $t_{current}$ and $t_{previous}$ represent, respectively, current time and

the time of the last order placement, in seconds. The effect of this adjustment is to make the exponential average more responsive if the order placement frequency is lower than 1 order per 10 seconds, and more inert if it is higher.

Given a suitable choice of β, the difference between the current price and the exponential average of past prices is a valuable indicator of current price dynamics. Namely, when the price is growing, the average is "too slow to catch up," and the difference is positive. When the price is falling, the average is too slow to decrease, yielding a negative difference. It is hoped that this quantity is informative enough to allow wise trading choices on the agent's part.

Finally, a note is due on the training procedure. Each of the strategies compared below was trained on 250 historical simulations, each encompassing over 15000 time steps, for a total of nearly 4 million Bellman backups. This amount of training effort was deemed to provide the agent with sufficient experience. Each simulation involved SOBI as the agent's only opponent. The trading days were a random mix of trading days in October 2003, the same for all strategies, but distinct from the balanced, handpicked collection of Appendix A on which their performance was compared.

2.4 RL1

Every strategy surveyed uses a reward function based on the agent's *present value*, a running measure of its profit and loss. The first strategy defines the agent's present value as its cash holdings plus its shares apprised at the last price. The state-action space includes share holdings, the difference between last price and the exponential average of past prices, and volume to purchase.

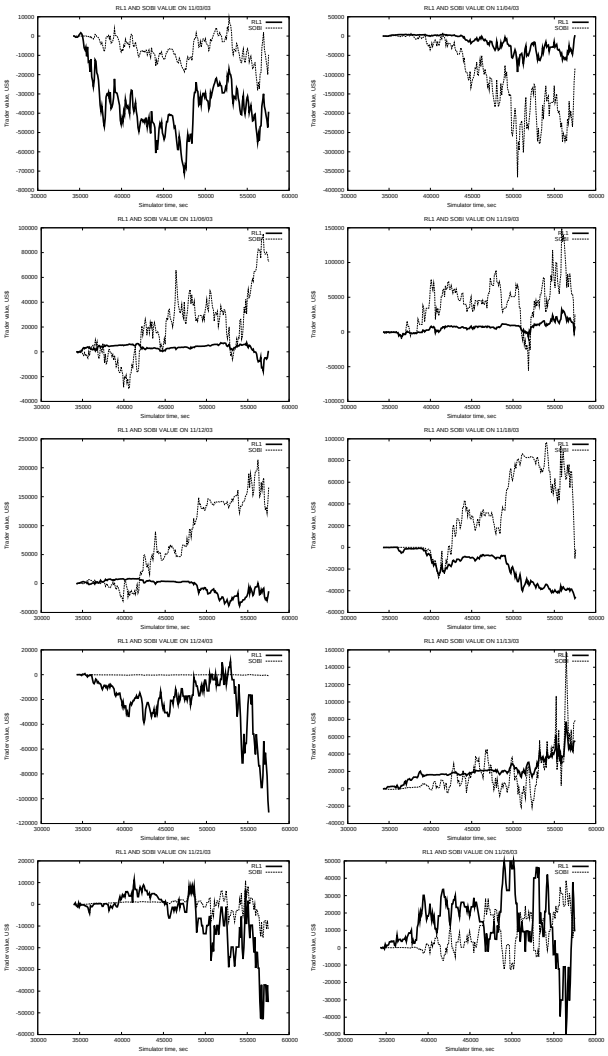


Figure 2: Comparison of RL1 and SOBI

Figure 2 shows the performance of this strategy against SOBI. RL1's value is plotted in a thick line. RL1 and SOBI exhibit comparably unsatisfactory performance on the two days with monotonic price behavior (top row), with SOBI winning when the price is monotonically increasing and RL1 winning when the price is monotonically decreasing. In either case, both agents end up with a negative value in the end. On days with substantial fluctuation in price (second row), RL1 performs better than in the monotonic scenario, and even manages to maintain a positive value for a while, but is still outperformed by SOBI. With one exception, the days with mixed and zigzag price behavior (last six graphs) are unprofitable for both traders; SOBI clearly comes out on top in this category as well. In sum, RL1 and SOBI seem to exhibit roughly comparable performance un-

der certain conditions, with SOBI performing better most cases.

While RL1's poor performance on days with mixed behavior is pardonable, it is particularly disturbing to see it perform so poorly on days with monotonic price behavior (in a sense, the easiest price trend). The results are all the more unexpected given that the price difference parameter was included in the RL framework specifically for the purpose of quickly detecting such price trends. A plausible flaw is the reward function itself. Because shares are valued at the last price, the agent is often punished even when it makes the right trading decision. Specifically, if the price exhibits a general upward trend, which is reflected in the positive price parameter, but briefly falls before picking up again (as is typical), the agent receives negative reward for choosing to buy shares. In other words, the reward function is too noisy to allow for the development of a meaningful strategy.

2.5 RL2

To eliminate the effect of the noise in present value calculations due to the jagged profile of the last price graph, this second strategy evaluates the shares at the exponential average of past prices. As Figure 1 reveals, the exponential average curve is much smoother than the last price graph, potentially yielding a more consistent reward function.

Figure 3 shows RL2's performance against SOBI. To avoid clutter, the graphs omit the SOBI curves which are much the same, and instead replicate RL1's curve from Figure 2 for easy comparison. RL2's value is shown in thick lines. The performance improvement from RL1 to RL2 is obvious: with the exception of two days (mixed and zigzag price behavior), RL2 outperforms RL1. The performance boost is apparently due to the more consistent and informative reward function. However, it is still unclear why RL2 does so poorly on days with monotonic price behavior: after all, the price difference parameter gives a clear indication of price dynamics in this case, and the reward function is much less noisy than RL1's. The reason for this phenomenon becomes evident upon a closer examination of RL2's reward function that values shares at the price average. When the price

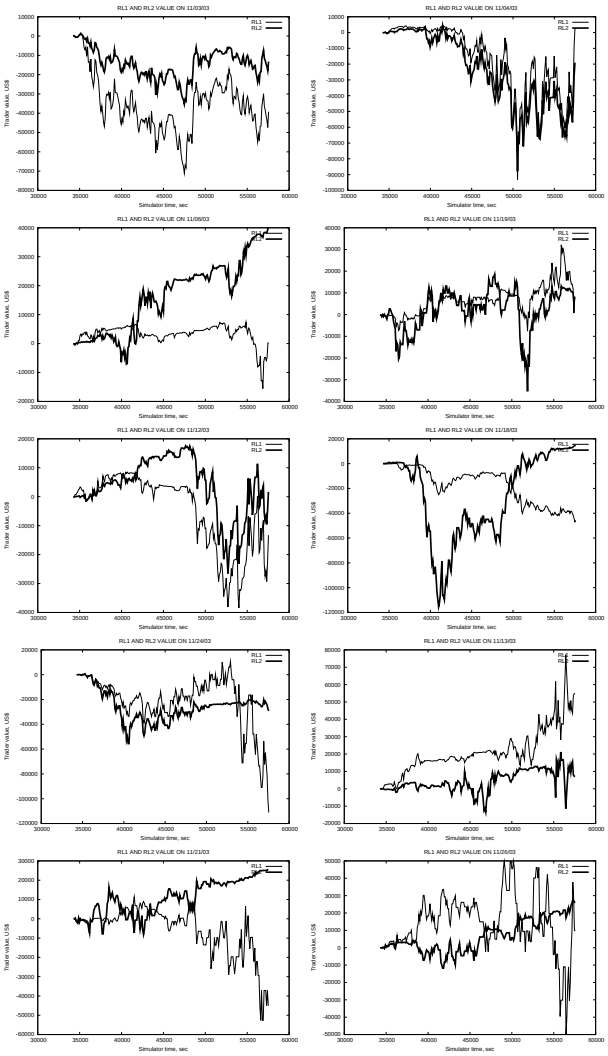


Figure 3: Comparison of RL1 and RL2

is going up, the price average is lagging behind the last price. Therefore, if the agent buys shares, it will actually receive negative reward because its decrease in cash will be proportional to the last price, but its increase in value due to a larger share position will be proportional to the (smaller) price average. An analogous argument applies to the scenario in which the price is steadily falling. In effect, the re-

ward function encourages "contrarian" behavior. If the price exhibits a monotonic trend, the agent will continuously lose value... while accumulating positive reward! Although RL2's reward function in an improvement on RL1's, it still leads to poor policies under certain market conditions.

2.6 RL3

While the first two strategies explored the effect of different formulations of the reward function on the quality of the policy, the final strategy is designed to assess the importance of share position as a state space variable. Specifically, RL3's state space is limited to the price parameter. The reward function is identical to RL1's, with shares valued at last price. Share position certainly seems to be a key determinant of optimal behavior along with a history of past market performance, and excluding share holdings from the state space is highly unlikely to yield near-optimal performance. However, the benefits of a simpler task formulation may well allow the development of a policy superior to RL1 and RL2's given the same amount of training experience.

Figure 4 plots the value of RL3 in simulations against SOBI. The SOBI curve is omitted, and RL1 and RL2's performance is plotted instead using the data from previous runs. RL3's value is plotted with thick lines. The graphs reveal the surprising result that excluding a seemingly essential variable from the state space leads to a policy of comparable quality. In fact, on 4 of the 10 days RL3 substantially outperforms both RL1 and RL2. Most notably, RL3 performs well on the days with monotonic price behavior that spell doom for the first two strategies. On two days RL3 loses to both RL1 and RL2, and exhibits roughly comparable performance on the remaining days.

The outcome of this comparison are indeed puzzling. The only obvious explanation is that all three strategies received the same training time allotment, although RL1 and RL2 were operating in a much more complex state-action space. The hypothesis, then, is that RL1 and RL2 did not have enough time to converge. It would be instructive to validate this claim in future work.

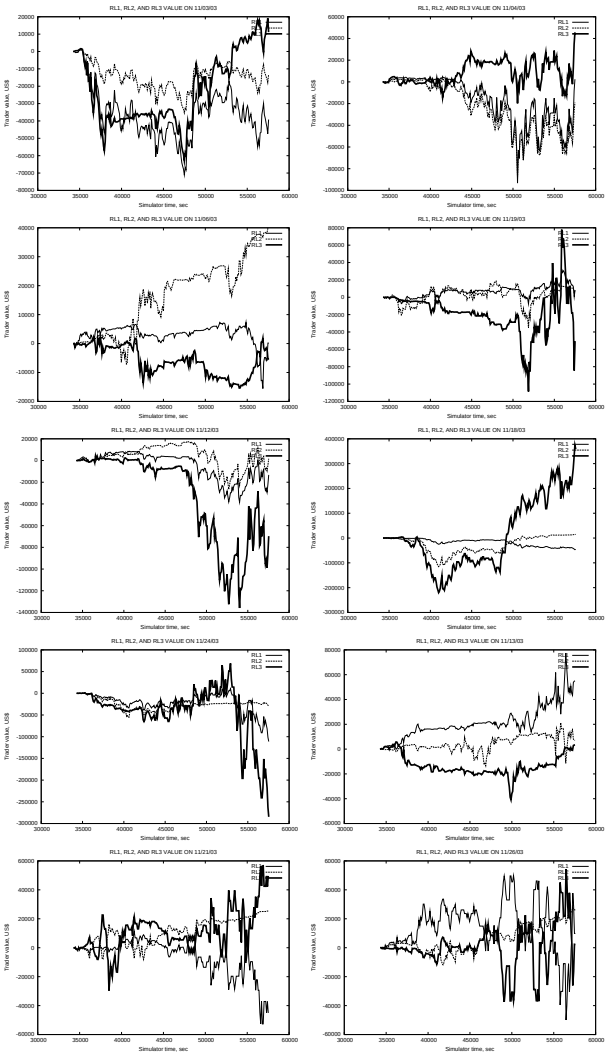


Figure 4: Comparison of RL1, RL2, and RL3

2.7 Evaluation

In summary, the proposed formulation of the trading problem as a reinforcement learning task yields somewhat discouraging results, in part due to the difficulties of adapting RL methodology to this specific domain. A major difficulty is due to the fact that the agent's action (order placement) may not have

any effect (i.e., the order may not be matched) until several time steps in the future. The reward function is oblivious to this fact, attributing any change in present value, which may well be due to random price fluctuation, to the last action taken. This misattribution of reward is likely to present a great impediment to learning.

Another difficulty with applying RL to stock trading stems from the fact that the learning mechanism blindly develops a best-response policy under the training conditions, which may not represent actual economies in which the agent will be deployed. Special care must be exercised to ensure that the agent's learning environment is sufficiently representative of future economies, a nontrivial challenge.

3 A Hill-climbing Approach

Unlike reinforcement learning, this approach represents a fixed strategy that constructs an explicit model of market dynamics and places orders accordingly. Specifically, linear regression is used to predict stock prices in the near future. Very roughly, the strategy is as follows. If the price is rising (i.e., the slope of the regression line is positive), the agent places buy orders, confident that it will be able to sell the purchased shares back at a higher price. If, on the other hand, the price is falling, the agent will place sell orders. In either case, the agent attempts to unwind its share position just before the price starts to drop (if it is currently on the rise) or just before the price starts to rise (if it is currently on the decline).

3.1 Strategy Design

The hill-climbing (HC) approach based on price prediction is best illustrated through an example. Figure 5 shows the Island last price on November 7, 2003. Figure 6 gives a corresponding graph of the S and SS curves[1], obtained as follows.

- The value $S(t)$ of the S curve at time t is the slope of the linear regression line computed using the price data for the past hour, i.e., for the time interval $[t-3600, t]$, where t is expressed in seconds. The length of the time interval presents a trade-off between currency (shorter time intervals generate S curves that are more responsive to the fluctuations in price, with the effect that the agent is able to quickly detect changing price trends) and stability (longer time intervals generate S curves that are more "inert" and less susceptible to random fluctuations in price). The interval width of an hour was chosen as a nice balance of these desirable characteristics. The key property of the S curve is that it grows, falls, and reaches local minima and maxima at the same time as the actual price graph. The purpose of the S curve is to distill growth and decrease information from the price graph, detecting genuine long-term price trends and ignoring short-term random price fluctuations.

- The value $SS(t)$ of the S curve at time t is the slope of the linear regression line computed using the S curve data for the past 400 seconds, i.e., over the time interval $[t-400, t]$. The width of the time interval over which the regression line is computed offers the same trade-off between responsiveness and stability; the value of 400 seconds seems to offer a good balance. The key property of the SS curve is that it is above the x-axis whenever the S curve is exhibits growth, and below the x-axis whenever the S curve is on the decline. Therefore, the $SS(t)$ value changes sign whenever the S curve reaches a local extremum and the price trend is reversed. The purpose of the $SS(t)$ is to alert the agent when the price trend is reversed.

In summary, by carefully studying Figures 5 and 6, one can see that the time intervals during which the $S(t)$ curve and the actual price curve grow coincide, as do the time intervals during which the two curves decline. Moreover, the $SS(t)$ curve changes sign whenever the current price trend is reversed. The signs of the $S(t)$ and $SS(t)$ curves induce the following categorization of possible price dynamics.

1. $S(t) > 0$ and $SS(t) > 0$: the price is growing at an increasing rate. This corresponds to a safe

[1] S is an abbreviation of "slope"; SS is an abbreviation of "slope of the slope."

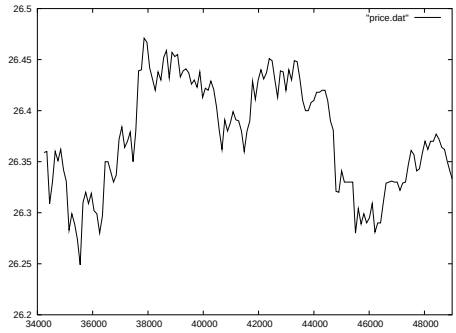


Figure 5: Island last price on November 7, 2003

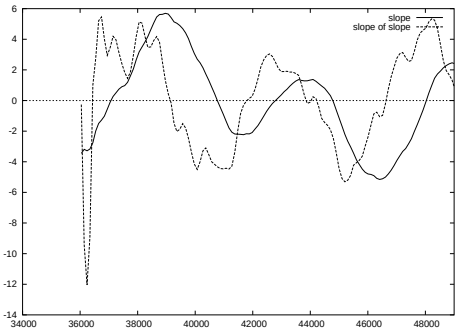


Figure 6: S and SS curves corresponding to the price graph in Figure 5

time to purchase shares with an eye to their subsequent liquidation at a then higher price.

2. $S(t) < 0$ and $SS(t) < 0$: the price is falling at an increasing rate. The market conditions favor the sale of shares because their will be readily bought back later at a lower price.

3. $S(t) > 0$ and $SS(t) < 0$: the price continues to grow, but the rate of growth is decreasing, suggesting a likely reversal in the price trend. This situation prescribes complete and immediate liquidation of all owned shares: the price is unlikely to continue growing much longer, and furthermore, the shares may soon start to depreciate.

4. $S(t) < 0$ and $SS(t) > 0$: the price continues to decrease, but the rate of decrease is shrinking, suggesting a likely reversion to growth in the near future. In this scenario, it is safest to buy back all owed shares since the price is unlikely to continue falling much longer, and what's more, the shares may start to appreciate soon, causing the trader's debt to balloon.

The above categorization of market conditions serves as the basis for the hill-climbing strategy. The ensuing three sections discuss variations of this basic design.

3.2 HC1: Basic Strategy

This section evaluates the performance of the basic HC strategy described in Section 3.1. In addition to the routine calculation of $S(t)$ and $SS(t)$ during each order placement cycle, the trader's actions under this strategy are as follows:

- $S(t) > 0$ and $SS(t) > 0$: place a buy order for 75 shares at the price $p_b + \Delta$, where p_b is the price of the top (most competitive) order in the buy book and $\Delta = \$0.001$ is the price quantum. This pricing scheme ensures that the order being placed will appear at the top of the buy queue and will therefore be likely to be matched soon. The volume setting of 75 shares leads to share positions as large as 150000 shares, serving as a rather loose leash. Further increasing the volume may complicate subsequent liquidation for two reasons. First of all, there may not be enough volume in the sell book to accommodate the trader's supply of shares at a reasonable price. Second, PXS does not currently support withdrawal of partially executed orders, which may present a substantial problem during liquidation if the purchase volume is increased far beyond 75 shares (see below).

- $S(t) < 0$ and $SS(t) < 0$: place a sell order for 75 shares at the price $p_s - \Delta$, where p_s is the price of the top (most competitive) order in the sell book and $\Delta = \$0.001$ is the price quantum.

- $S(t) > 0$ and $SS(t) < 0$: withdraw all pending buy orders and partially liquidate the owned

shares by placing a sell order for s_b shares at a price of p_b, where s_b and p_b are the volume and price of the top (most competitive) buy order. This liquidation scheme ensures rapid liquidation at a tolerable cost (only the most competitive buy order is matched).

- $S(t) < 0$ and $SS(t) > 0$: withdraw all pending sell orders and buy back some of the owed shares by placing a buy order for s_s shares at a price of p_s, where s_s and p_s are the volume and price of the top (most competitive) sell order.

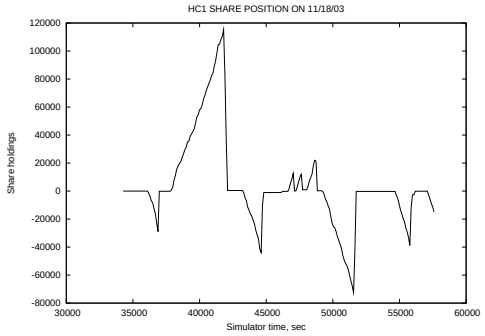


Figure 7: HC1 share holdings on November 18, 2003

Before comparing the performance of the HC1 and SOBI strategies, it is instructive to inspect the change in HC1's share position over the span of a typical trading day. The share position graph in Figure 7 corresponds to the price graph for 11/18/2003 in Appendix A. The graph reveals that the changes in HC1's share position closely correspond to the price behavior. Namely, the price was steadily growing through $t \approx 42000$, triggering uninterrupted buying on the part of the agent. The drop in price between $t \approx 42000$ and $t \approx 45000$ corresponds to the complete liquidation of the agent's share position and the subsequent selling period. After some random fluctuation through $t \approx 48000$, the price continues to drop, causing the agent to sell shares through the end of the simulation. Observe that the agent periodically unwinds its share position even when the price trend seems unchanged. This behavior is due to the imperfect nature of the predictions obtained using the SS curve. In this particular time interval, the SS curve seems to be too "sensitive" to price fluctuations, generating false alarms when in fact no trend reversal is taking place. However, "dulling" the SS curve's responsiveness to the market conditions would not allow the agent to react as quickly to a genuine trend reversal.

Figure 8 shows the performance of the HC1 and SOBI strategies over the span of trading days selected as a representative sample of typical market trends. The corresponding price graphs can be found in Appendix A.

The top two graphs correspond to days on which the price was steadily increasing (left) and decreasing (right) throughout the day. As expected, HC beats SOBI on these days by ensuring that it does not hold any significant positive share position when the price is declining or a negative share position when the price is increasing. The next row corresponds to days on which the price fluctuated greatly throughout the day, and SOBI apparently has a competitive edge because it does not rely on longer-term price trends. The next three graphs show days with zigzag price behavior. In one case, HC wins and finishes the day with a noteworthy positive balance; in the other two cases, HC loses to SOBI. The final three graphs correspond to mixed price behavior, with 1 win and 2 losses for HC. The fact that HC realizes the best value on 11/18/2003 jibes well with the intuition that HC should perform best on days with price trends of medium duration: shorter trends diminish the value of prediction, while longer trends often contain aberrations within a trend that trigger premature unwinding on the trader's part.

3.3 HC2: Cautious Order Pricing

This variation of the hill-climbing strategy aims to improve on the basic design through a less ambitious order pricing scheme. More specifically, when the trader purchases shares for resale or sells shares for future purchase under HC1, orders are always placed such that they appear at the top of the corresponding book. This approach ensures rapid execution of the orders but is not strictly supported by the price prediction model because of the disparity between the

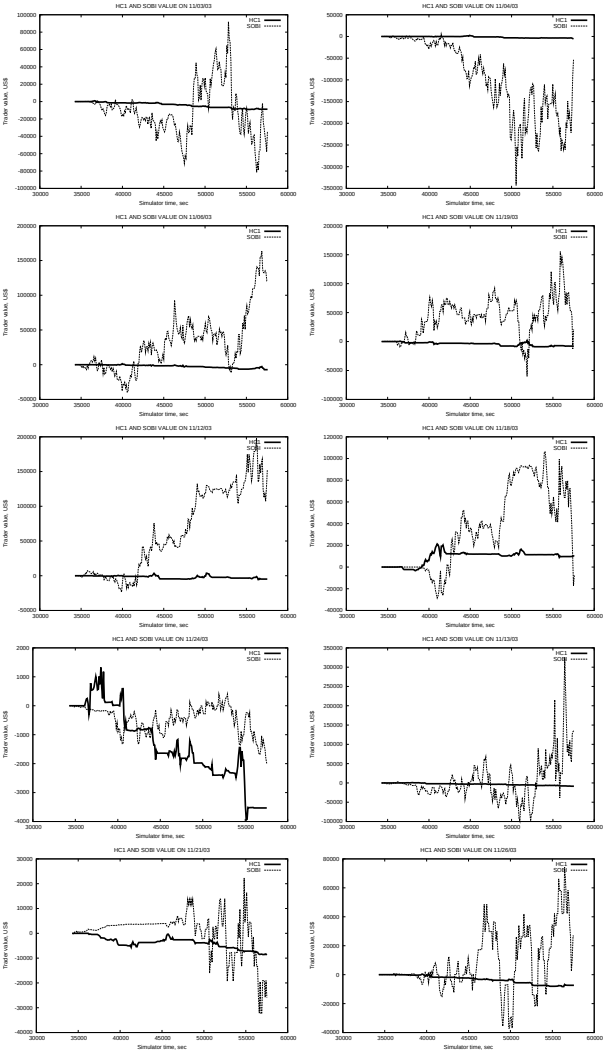


Figure 8: Comparison of HC1 and SOBI

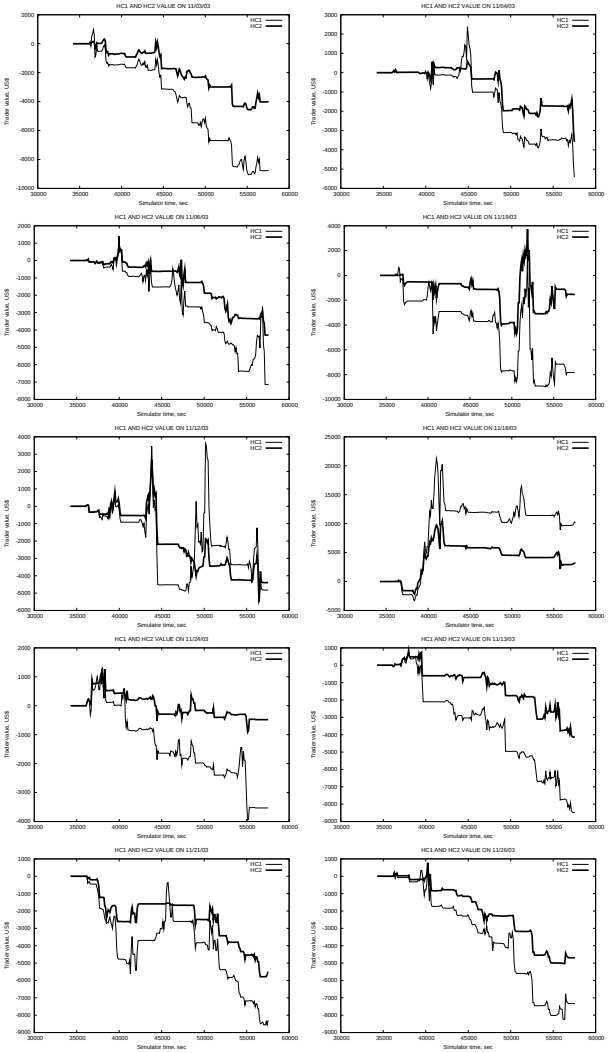


Figure 9: Comparison of HC1 and HC2

predicted price at the current moment and the price at which the order is placed.

In concrete terms, suppose the price is rising at an increasing rate and the trader is entering a buy order at the price p. Suppose further that, according to the latest linear regression model, the price at the current moment is predicted to be $\hat{p}$. While there is good reason to believe that the price will soon rise above $\hat{p}$, there is no guarantee that the price will rise above p if $p > \hat{p}$. The hypothesis underlying this modified hill-climbing scheme is that in such case the trader should place a buy order at $\min(p, \hat{p})$. While this modification may in some cases decrease the likelihood of matching, it gives us a greater assurance that the trader will not overpay for the shares it purchases with the intent to sell at a later time. Analogous rea-

soning suggests that when the price is falling at an increasing rate, the trader should sell shares at the price $\max(p, \hat{p})$. No other aspects of the original HC strategy, including the liquidation scheme, are altered by this modification.

The graphs in Figure 9 bear out the expediency of this modification. Each graph was obtained by matching HC2 against SOBI. The SOBI curve is omitted, and the HC1 curve from Figure 8 added to the graph for comparison purposes. HC2 performance is plotted with thick lines. The graphs indicate that HC2 outperforms HC1 in almost any market conditions, with the exception of the zigzag price behavior on 11/18/2003, HC1's most successful day. The competitive advantage of HC2 is especially pronounced when market conditions do not favor a linear prediction model (e.g., the mixed price behavior on 11/21/2003, 11/24/2003, and 11/26/2003) and HC1 loses a considerable amount.

3.4 HC3: Periodic Liquidation

This final modification to the hill-climbing approach is based on the recognition that the success of the original strategy hinges on too many contingencies over which the trader has no control. In particular, the price prediction module may not give the trader a sufficiently early warning of a trend reversal to allow for efficient, low-cost unwinding. Moreover, even with an advance warning, the other participating traders may be disinclined to supply shares to the trader (if the price has been falling and the trader has been selling shares) or to buy shares from the trader (if the price has been rising and the trader has been buying shares), causing the trader to pay an exuberant price for unwinding its share position. Instead of pinning all hopes on last-minute trading, the periodic liquidation approach prescribes unwinding the share position at regular intervals. This approach will certainly yield lower profits than the best possible profits obtained using last-minute unwinding but will shield the trader to some extent from the unforeseen outcomes of delayed liquidation.

Specifically, in this approach the trader completely unwinds its position after an hour of uninterrupted selling or buying of shares. The periodicity of unwinding (1 hour) was chosen on the grounds that a typical medium-term price trend lasts just over an hour, as can be verified by inspecting the price graphs in the appendix. This periodicity allows the trader to unwind its position early, thus avoiding the last-minute scramble and an uncooperative economy, but not so early as to make the gains of trend-based buying and selling negligible.

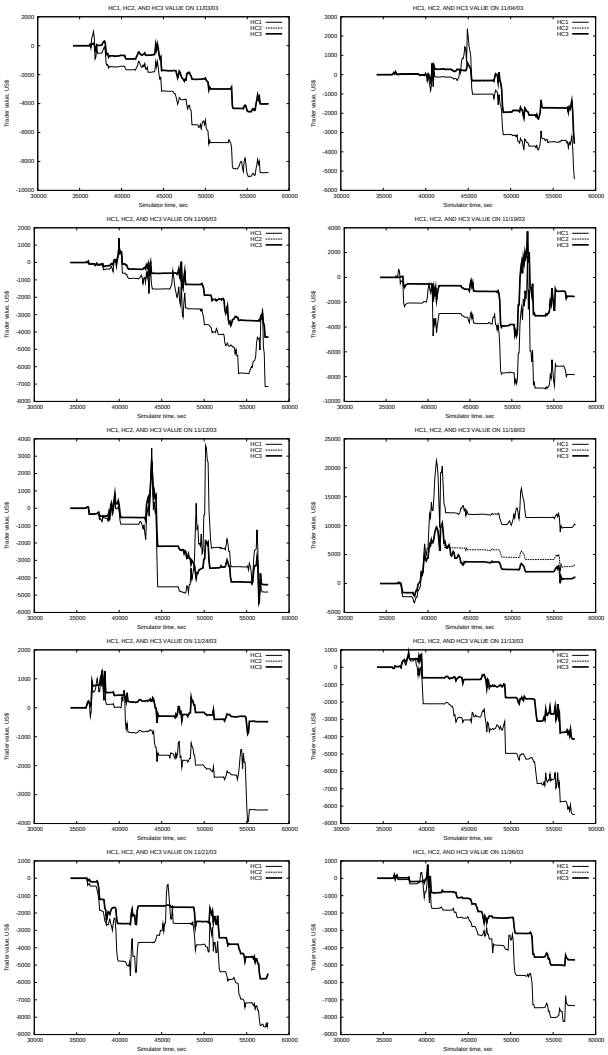


Figure 10: Comparison of HC1, HC2, and HC3

Figure 10 was obtained by running HC3 and SOBI in a joint experiment. HC3's value is plotted with thick lines. The graphs, however, that HC3 is little improvement on HC2. The HC2 and HC3 strategies have virtually indistinguishable performance curves in all market conditions except for 11/18/2003, when HC2 slightly outperforms HC3. The observed performance of HC2 and HC3 is apparently due to the fact that HC2 almost never sells (or buys) shares without interruption for over an hour. The SS curve is sensitive enough to generate occasional false positives (i.e., diagnose the price behavior as a trend reversal when, in fact, the market is experiencing a brief aberration from the trend and the trend will shortly resume). As a result, HC2 "automatically" unwinds its position at more or less regular intervals, without a need for an explicit strategy rule to this effect.

3.5 Evaluation

Although the hill-climbing approach is founded on a seemingly sound idea that shares should be sold when the price is falling and bought when the price is rising, practical instantiations of this approach do not seem to yield much profit. In fact, the graphs reveal that it is typical for HC to steadily lose value throughout the day, albeit at a very slow rate.

On the one hand, the (relatively illiquid) economy created by the SOBI agent may not be representative of a typical liquid market; HC could thrive in a more favorable economy. Moreover, HC's profit and loss is not subject to the great variation observed in SOBI, suggesting that HC's success is contingent on the price dynamics to a much lesser extent. The latter property of the HC strategy makes it an appealing choice if the primary performance criterion is the Sharpe ratio, which emphasizes consistent profit and loss. Finally, HC does outperform the SOBI strategy in certain market conditions.

4 Market Making

As discussed above, the objective of the hill-climbing strategy is to look for long-term trends in price fluctuations, buying stock when the price is low and later selling stock when the price has gone up (and vice versa with the price going in the opposite direction). As such, the performance of the strategy is highly dependent on the price dynamics of a particular trading day. As a consequence, one might expect significant variance in profit over a span of several days. If more consistency is desired, an approach based on market making (MM) may be more useful. Unlike the hill-climbing strategy, the MM strategy capitalizes on small fluctuations rather than long-term trends and is likely to produce a smaller variance in profit.

4.1 Strategy Design

This final approach to the stock-trading problem combines the regression-based price prediction model presented in Section 3 with elements of market making. Specifically, under this strategy buys stock when the price is increasing at an increasing rate and sells stock when the price is decreasing at an increasing rate. However, rather than wait for a trend reversal to unwind the accumulated share position, the agent always places buy and sell orders in pairs. When the price is increasing at an increasing rate, the agent places a buy order at price p (calculated as the price of the top order in the buy book minus a price quantum) and immediately places a sell order at price $p + \Delta$, confident that the latter will be matched shortly when the price has gone up enough. The Δ parameter is the per-share profit the agent expects to make on this transaction. The specific implementation of this approach below use $\Delta = 0.01$ as a sufficiently profitable yet safe choice. The situation is symmetric when the price is decreasing at an increasing rate. Finally, the agent takes no action during the periods designated as "price reversal" by the prediction module (with price increasing or decreasing at a decreasing rate): since the orders are placed in pairs at what is deemed a "safe" time, no additional effort is called for to unwind the share position.

4.2 MM1: The Basic Approach

The MM1 strategy implements the simplest version of the approach just described. In particular, no ac-

tion is taken to monitor share holdings during price trend reversals. Figure 11 plots the performance of this approach against SOBI, with MM1's performance curve drawn with thick lines.

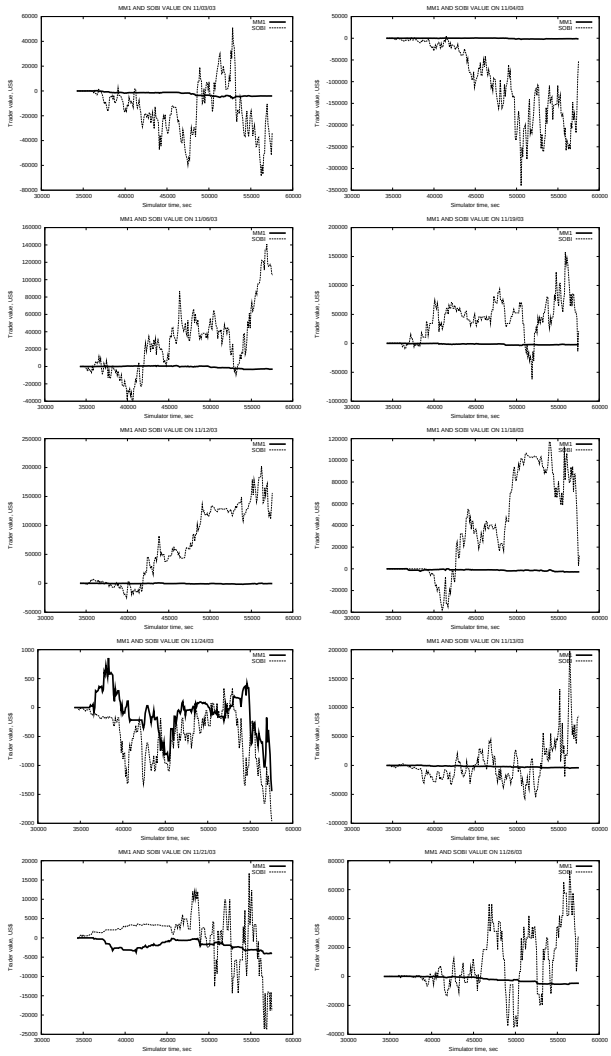


Figure 11: Comparison of MM1 and SOBI

The graphs show that, far from making a consistent profit regardless of the economy, the strategy steadily loses value under all market conditions. On six of the trading days, SOBI's significantly outperforms MM1. The reason for the consistently poor performance of MM1 is its naïve order placement policy. Specifically, the strategy always places buy and sell orders in pairs. When the price is growing, the strategy places a sell order under the assumption that the corresponding buy order will be matched. However, this assumption is hardly justified. Instead, MM1's competitors can ignore its buy orders and match its relatively cheap sell orders—a boon when the price is on the rise. In essence, MM1 places itself in an extremely vulnerable position by placing a sell order *before* its buy order is matched. An analogous situation arises when the price is decreasing.

4.3 MM2: Conditional Order Placement

MM2 eliminates the flaw in MM1's order placement policy. Specifically, it still places a buy order when the price is increasing at an increasing rate, and a sell order when the price is decreasing at a decreasing rate, but does not place the other order of the pair until the first order is matched. The latter is accomplished by maintaining a list of *conditional* buy and sell orders, i.e., orders whose placement is conditional on the successful matching of another order. During each order placement cycle, the strategy additionally requests a list of its unmatched orders from the simulator and places any order whose counterpart no longer appears in the books.

Figure 12 plots the performance of MM2 in a joint simulation with SOBI. The SOBI curve is omitted, and the MM1 curve is replicated from Figure 11 for comparison purposes. The graphs show a dramatic performance improvement: MM2 significantly outperforms MM1 on 8 days and performs the same on the other 2 days. The latter 2 days correspond to zigzag price behavior, an apparently challenging economy for the model used; a plausible explanation is offered in the next section. It is remarkable that MM2 achieves a consistent positive value 70% of the time, suggesting that the strategy's performance is indeed largely independent of the price dynamics.

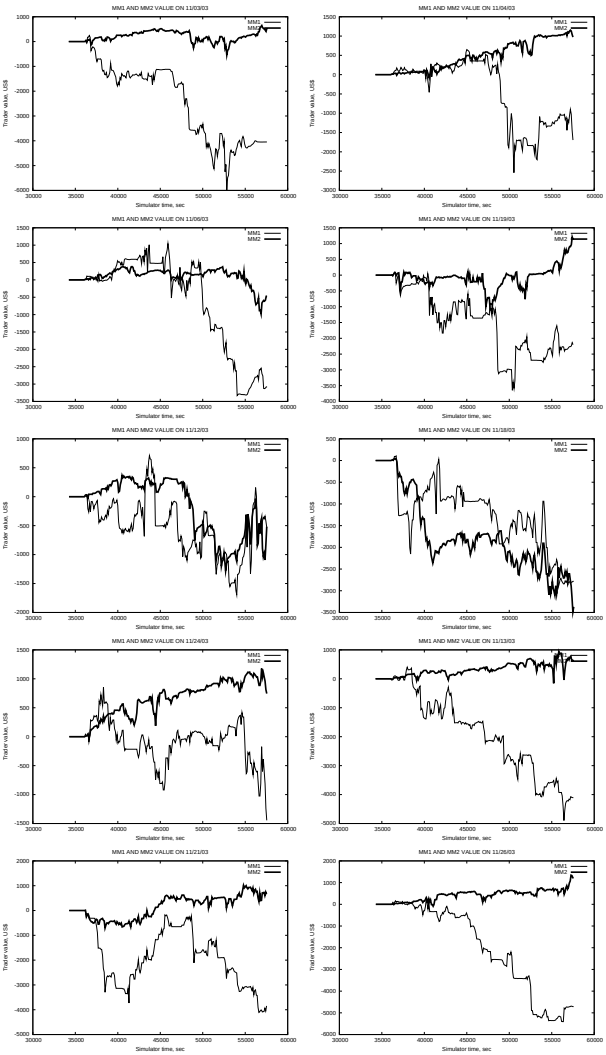


Figure 12: Comparison of MM1 and MM2

4.4 MM3: Further Restrictions on Order Placement

A closer examination of MM2 suggests that there may be room for improvement. First, it seems useful to reintroduce a limited form of unwinding into the model. In a conceivable scenario, an order is placed and matched, and its counterpart order is placed next. However, the latter order fails to get matched, creating an excess or deficiency in the agent's ideally zero share holdings. If the price changes unfavorably (i.e., rises when the agent owes shares or falls when the agent owns shares), this share imbalance will incur a substantial cost. This phenomenon is likely responsible for MM2's poor performance on days with zigzag price behavior: the price peaks and then continues to drop through the end of the day, with the likely consequence that many of the conditional orders placed shortly before the peak will not be matched in time. An analogous scenario occurs when the price reaches a global minimum and starts growing from that point on. MM3 solves this problem by buying back any sold shares whose corresponding conditional buy orders failed to be matched within 5 minutes of placement, and likewise for any purchased shares. The 5 minute time limit seems appropriate given that the strategy places very competitively priced conditional orders. Moreover, a time limit so short will ensure that unfavorable price changes are weathered at a low cost.

Another improvement on MM2 implemented in MM3 prevents the agent from placing a "primary" (non-conditional) order if there are any unmatched conditional orders in the corresponding book. The rationale behind this modification is as follows. Because primary orders are always priced such that they appear at the top of the book, their placement interferes with the successful matching of the conditional orders already in the book. The matching of conditional orders is absolutely vital to the success of the agent. With this in mind, MM3 refrains from submitting any primary orders to a book that contains unmatched conditional orders.

MM3 implements a number of other modifications. First of all, instead of trading a fixed amount of shares per order, it decreases the trade size of a primary buy order proportional to the unmatched conditional sell volume in the books, and likewise for a primary sell order. This change allows the agent to adjust its trading activity based on how successful its has recently been at unwinding accumulated negative or positive share holdings for profit. Second, when the price prediction module signals a trend reversal, the agent withdraws all unmatched primary

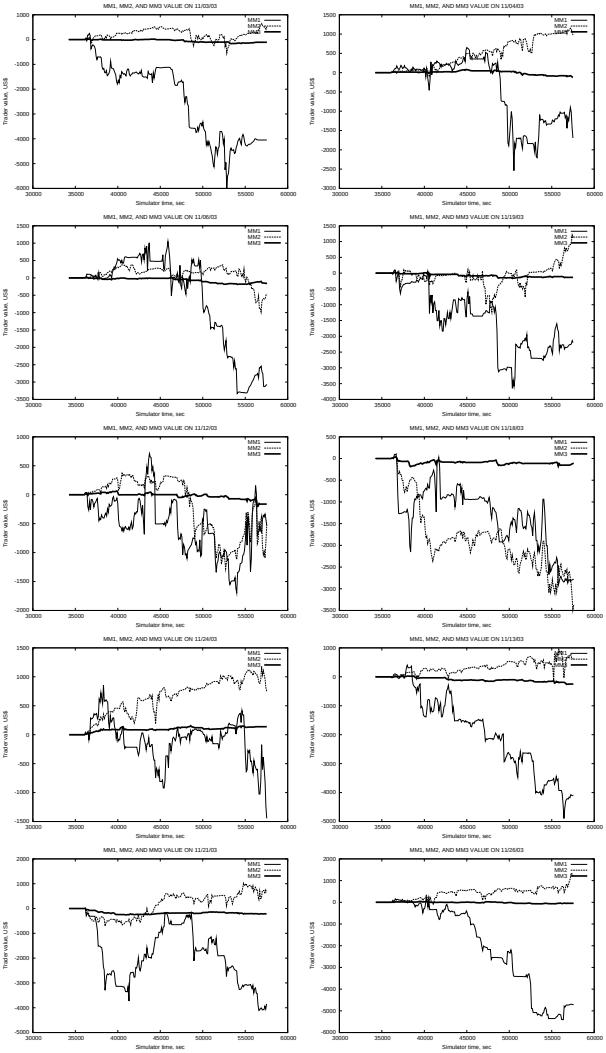


Figure 13: Comparison of MM1, MM2, and MM3

orders, realizing that it would not be able to have their conditional counterparts matched later, a phenomenon discussed above. Finally, to compensate for the revenue lost due to the reduced volume of trades, the agent submits every conditional buy order at the minimum of its corresponding price under MM2 and the price of the top order in the sell book; and likewise for the conditional sell orders. This change allows the agent to extract more profit per share at the cost of a minimal reduction in the likelihood of the conditional order's being matched.

Figure 13 plots the performance of MM3 (thick line) against SOBI, with MM1 and MM2 curves replicated for reference from Figures 11 and 12. The graphs clearly indicate that the numerous restrictions and precautions incorporated into MM3 reduce its trading activity to the point where the agent barely trades at all, maintaining a near-zero value under nearly any market conditions. Given that the modifications described above are motivated by sound reasoning, it would be an interesting research idea to identify a subset of the changes, as well as the parameter settings (e.g., the 5-minute matching time limit for a conditional order), that optimize the agent's performance.

4.5 Evaluation

Based on the performance of MM2, the most successful of the market-making implementations presented, this approach is very promising. In particular, neither the reinforcement learning nor hill-climbing approach come close to rivaling MM2's profit consistency. An important extension for MM2 to be viable in practice is an adaptive mechanism for determining the size of trades and the time limit for a conditional order to match. These parameters are likely to have different optimal settings in different economies and under different market conditions, and a fixed setting is hardly appropriate. Moreover, there is an inherent trade-off between the size of the trades and Δ, the requested profit per share. If the agent trades high volume of shares, it will have to accept narrow profit margins (or else see its conditional orders unmatched); if the agent trades little, it can afford to extract a more ambitious profit per share. A way of optimizing this trade-off would be a valuable addition to the strategy.

5 Comparative Analysis

To evaluate the comparative performance of the presented approaches, I selected the best implementa-

tion is each category (RL3[2], HC2, and MM2) to participate in joint simulation with SOBI.

This time, each strategy was allowed to run until 3 p.m., at which point control was turned over to a position-unwinding module. The position-unwinding approach used is extremely straightforward. The agent starts by withdrawing all its unmatched orders. Then, if the agent owes any shares, it will place a buy order, one per order placement cycle, for s shares at price p, where s and p are the volume and price of the top order in the sell book. The liquidation of any owned shares proceeded likewise. This unwinding method allows for rapid unwinding at a tolerable cost. The experiments described below validate the effectiveness of this procedure: all four agents finished the day with zero share holdings.

A strategy's score on a given trading day was total profit and loss (i.e., value) at the end of the day *plus* total rebate value (computed as \$0.002 per share that added liquidity to the simulator) and *minus* total fee value (computed as \$0.003 per share that removed liquidity from the simulator). The final rank of a strategy was based on its Sharpe ratio, defined as its aggregate score over the 10 days divided by the standard deviation of its 10 scores.

Due to clutter and scaling problems, it is difficult to display the results of the joint simulation in graph form. Instead, a tabular format is more appropriate. Table 1 contains performance data for every strategy and every trading day. As can be readily seen, RL3 and HC2 were largely unprofitable, finishing with a negative score on 8 of the days. RL3's performance was particularly poor, as the large negative scores indicate. MM2 and SOBI, on the other hand, are consistently winning, finishing with a positive balance on 7 of the days. Of the four strategies, SOBI's scores are the most impressive.

Table 2 shows the final rankings in this joint simulation, as determined by the Sharpe ratios. It is noteworthy that MM2, generating profits that are a

[2]RL2 and RL3 were close competitors. I chose RL3 over RL2 because the former policy was presumably closer to convergence, as discussed in Section 2.6, and its performance in a competition against other strategies would be a more meaningful indicator of how successful a reinforcement-learning formulation could be at this task.

	Day	Value	Fees	Rbts	Score
RL3	03	-9368	4542	6597	-7314
	04	-42491	3784	5563	-40712
	06	-12163	4530	5713	-10980
	12	-162219	3989	6029	-160178
	13	-22217	4493	5730	-20981
	18	-208233	6358	5314	-209277
	19	-20862	6775	4890	-22747
	21	29952	6406	4799	28345
	24	144	5187	4051	-992
	26	20782	6078	4595	19299
HC2	03	-2334	705	380	-2659
	04	-1366	654	397	-1623
	06	-1917	632	429	-2119
	12	-882	689	412	-1159
	13	-186	647	403	-430
	18	6338	661	368	6045
	19	-3003	957	491	-3469
	21	-3302	786	411	-3677
	24	352	677	416	90
	26	-4384	772	380	-4776
MM2	03	672	494	514	692
	04	925	329	491	1087
	06	-115	471	573	-13
	12	-1486	433	598	-1321
	13	611	469	542	684
	18	-1592	388	679	-1300
	19	-78	392	579	108
	21	615	540	660	735
	24	902	381	561	1081
	26	195	456	520	259
SOBI	03	-1409	159	2119	550
	04	-27309	883	4193	-23999
	06	50749	161	844	51432
	12	97779	901	2611	99489
	13	42351	193	930	43088
	18	75660	2049	1959	75569
	19	14386	717	1880	15550
	21	-6518	201	502	-6216
	24	-2347	4	63	-2289
	26	21887	358	767	22295

Table 1: Performance of RL3, HC2, MM2, and SOBI in the joint simulation

tiny fraction of SOBI's, finished with a Sharpe ratio quite close to SOBI's. This fact is due to the emphasis on consistency built into the Sharpe ratio. An important lesson to be learned from this competition is that if the Sharpe ratio is the primary criterion, large profits are not strictly necessary for placing in the top ranks; a consistent strategy that generates small profits would be a strong contender, too.

The rankings offer valuable insights into the performance of the three approaches developed in this project. The poor performance of RL3 in a joint simulation may reflect that fact that its training experience did not incorporate key features of the joint economy. The hill-climbing approach, too, performs very poorly. The market-making strategy, on the other hand, thrives in this new economy. Although it generates relatively little profit, its earnings are quite consistent and not much affected by the market performance. Of the three strategies surveyed in this paper, the market making clearly emerges as the strategy of choice.

6 Conclusions and Future Work

This report documents the development of an automated stock-trading application. Three approaches are presented, evaluated individually against a fixed opponent strategy, and analyzed comparatively. Suggestions for future work are put forward in the concluding subsections of each approach. This study confirms that automated stock trading is a difficult problem, with reasonable heuristics often leading to marginal performance and small-profit strategies proving highly competitive, according to important metrics, with very profitable strategies. While the area of stock trading has received much attention in the past, the unique opportunities and challenges of up-to-date order book information and of electronic share exchanges, as exemplified in part by the presented approaches, merit further study.

Rank	Strategy	Sharpe Ratio
1	SOBI	7.014
2	MM2	2.290
3	HC2	-4.573
4	RL3	-5.428

Table 2: Final rankings in the joint simulation

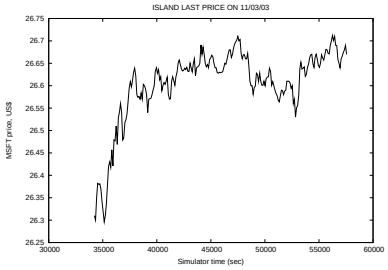
References

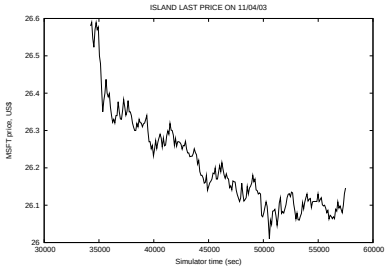
[1] Kearns, M., & Ortiz, L. *The Penn-Lehman Automated Trading Project.* To appear, IEEE Intelligent Systems, 2003.

[2] Sutton, R., & Barto, A. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, Mass.: 1998.

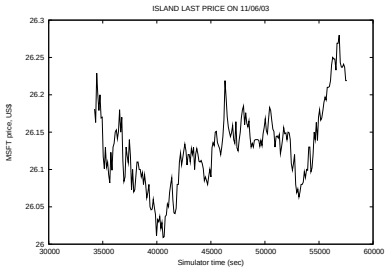
Appendix A: MSFT Last Price by Date

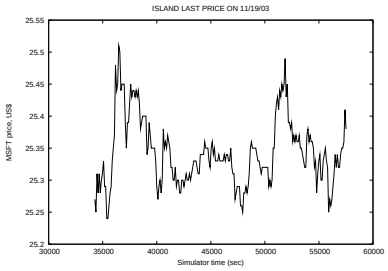
Monotonic decrease/increase



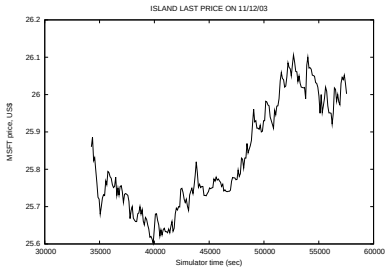


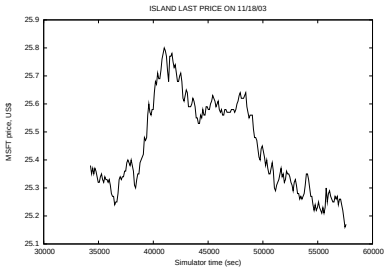
Substantial fluctuation

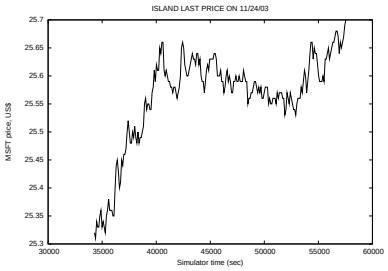




Zigzag behavior







Mixed behavior