Universal Reverser for the PXS Local Market

Gurushyam Hariharan

1. Introduction

In this paper I present an intuitive approach to automated trading tested in the PLAT simulation environment. The basis of the approach is to explore an arbitrage opportunity that exists because of the simultaneous presence of two separate, yet interdependent markets viz. 1) the Penn Exchange Simulator (PXS) market and 2) the Electronic Communication Network (ECN) (here Island). My strategy is to capture the difference in the demand and supply in the two markets and explore the gap between the two. The way it explores the gap is by playing against the cumulative mood of the local market. For this reason I call my agent the universal reverser for a local market.

The remainder of the report is organized as follows. Section 2 of the paper discusses the motivation and hypothesis of the strategy. The strategy in its most basic form is presented in section 3. The subsequent section discusses some evaluations with experimental analyses and reasoning. This is followed by a note on related and future work.

2. Motivation and Hypothesis

PXS aims for providing a test bed for simulating virtual orders and matching them seamlessly with the real world orders. In doing so, PXS creates a local market that is separate from that of the ECN. At every clock tick of the server, it tries to bridge a gap that is developed between the two markets. My hypothesis in the strategy is that the local market of PXS is very small as compared to the bigger market of the ECN. Hence, every local demand or supply that is created in the PXS market will be easily subsumed by the bigger ECN market.

In the discussions to follow I will denote supply and demand of a market as the characteristics of the market. The abbreviation UMR (Universal Market Reverser) will be used to denote the agent. To understand the differences in the characteristics that exist in the two markets, I present an overview of the working of that part of the simulator that creates the local market and matches it to the ECN.

At every tick, (virtual) orders are taken from the virtual clients. First an attempt is made to match them amongst themselves. If they are successfully matched, they are removed from the order book. If they are not, an attempt is made to match them with the real world order book. If they can be matched, the real world orders are removed from the PXS order book, else the virtual orders are inserted into the same. The PXS buy and sell books are processed in order. All matches between the buy and sell orders that cross are matched. The last price at which the sell and buy virtual orders are matched is the simulator last price for the present tick count. A similar last price also exists for the ECN. It is called the Island last price.

The simulator last price (SLP) and the Island last price (ILP) represent the contemporary characteristics of the two interdependent markets. If these were independent markets operating separately, I could buy shares from one of these markets whose last price is the least and sell them in the other market. The profit that I would make is absolute difference: *abs(ILP – SLP)* per share. Since there is a real time matching of orders in these markets, the above strategy is not possible in this domain.

One of the ways the difference in the two last prices could be exploited is if we hypothesize that the PXS market is very small as compared to that of the ECN. The hypothesis would mean that the two prices would tend to converge, and the hypothetical point of convergence would be the ECN price. Hence, difference in the two prices can be seen as an opportunity for an arbitrage. In plain words, if SLP is less than ILP we could see this as an opportunity to buy stocks in a local market at a price lower than the actual market price (read, the island price). The same stands for the opposite scenario.

3. The Basic Agent

The basic agent strategy can be represented by the following algorithm

```
While time permits {
        ILP = getIslandLastPrice()
        SLP = getSimulatorLastPrice()
        diff = ILP – SLP
        If (diff < 0)
                placeOrder(SELL, price, volume)
        else if (diff > 0)
                placeOrder(BUY, price, volume)
}
```

Strengths and weaknesses

The strategy would work well if the local market (the PXS market) trades shares in volumes less than that in the ECN. In this case, the ECN market can be seen to be a huge market that subsumes the smaller PXS market. Hence the market characteristics of the smaller PXS market would be driven by the ECN market.

There are a number of parameters that can be adjusted for a good performance of the strategy. Experiments that deals with different combinations of the same are be discussed in section [4].

The strategy would work well if there is an appreciable difference between the ILP and SLP. For example, if the SLP is $28, and the ILP is $30. If I buy one share at the SLP, I have made a profit of 30-28= $2. On the other hand, if the local market trades very close to the island price, no arbitrage opportunity emerges. Hence the agent doesn’t make profit.

In a way, the agent tries to cumulatively play against the local market strategies. If the cumulative mood is to buy, a demand is created, and the agent sells shares. On the other hand if the cumulative mood is to sell, an excessive supply is created and the agent buys shares. In this way, it seamlessly replicates all individual market reversal strategies. E.g. the above discussed scenario is precisely what is achieved by the Reverse strategy discussed in [3]. The only difference being, the latter plays against the expected market mood, whereas, UMR plays against the actual market mood.

On the same note, if the agents in the market are ambivalent, they do not create a deterministic demand or supply. In other words, they are unable to define a cumulative market mood effectively. The price difference could fluctuate in the positive and negative values with high frequencies. In such cases, UMR will not get useful information from the difference in prices and hence might not function effectively.

4. Experimental results and performance analysis

As mentioned in earlier sections, there are a number of optimizations that can be achieved with proper parameter tuning. Here, the methodology being adopted in tuning the parameters is pure experimentation and reasoning. A number of experiments were conducted to achieve optimizations. The following are results of the some salient experiments.

Each of the experiments is performed with 4 agents: two dummy agents (SOBI), one final agent submitted for the competition, and lastly the agent *lacking* the particular behavior being discussed. The dummies are included since the UMR based agents cannot trade on their own. They need a local market to be created. The dummies are assumed to create the required local market. In this way, I directly contrast the effectiveness of the behavior being discussed.

In any trading strategy, the foremost tuning parameters are 1) the volume of stocks being traded in each order, 2) value at which an order is placed.

4.1 The Price Parameter

The following are my salient graphs from some of my experiments to tune the price parameter.

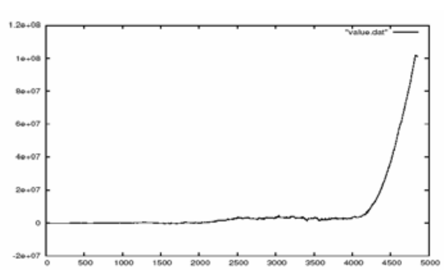


Figure 1: Value of the shares traded for the day

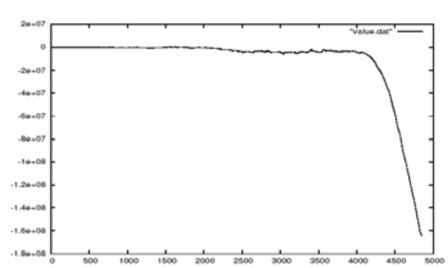


Figure 1: Value of the shares traded for the day

*Figure[1] The Delta effect: The performance of the final agent with the price parameter tuned is shown in the first graph. The second graph shows the agent with a constant Delta value. Clearly, the first trader wins by a margin of almost $ 2 * 10^8*

The price parameter can be fixed to be a constant value above or below the simulator price, depending on whether it is a buy or a sell order.

eg. buyorder (BUY, volume, currentPrice + Delta) and
eg. sellorder (SELL, volume, currentPrice - Delta)
with Delta = constant

Experiments show that, if Delta is made a function of the difference between the simulator and the island prices, the agent makes more profit. The following are the graphs comparing the performance of the agent with a constant Delta, and that with a Delta as a function of difference. The strategy being used in the competition is

eg. buyorder (BUY, volume, currentPrice + Delta) and
eg. sellorder (SELL, volume, currentPrice - Delta)
with Delta = 0.1 * abs(IslandPrice - SimulatorPrice)

The reasoning behind the better performance of the non-constant delta, is the following. Price of a share is best fixed with the help of the knowledge of demand and supply in the market. By making Delta a function of the difference in the price of island and the simulator markets, we are directly making it a function of the local demand and supply. Hence the agent incorporating the knowledge of the characteristics of the market performs better than its counterpart.

4.2 The Withdrawal Effect

It is evident from the strategy specification that the decision at one tick of the simulator is not valid at another tick. This observation is incorporated in the agent as follows. Whenever a new decision is made (i.e., at every tick of the simulator), all the previous orders are cancelled. In doing this, the effects of the past decisions are removed from the present state of the agent.

The following are the graphs comparing an agent that shows the above behavior, with that which lacks the same.

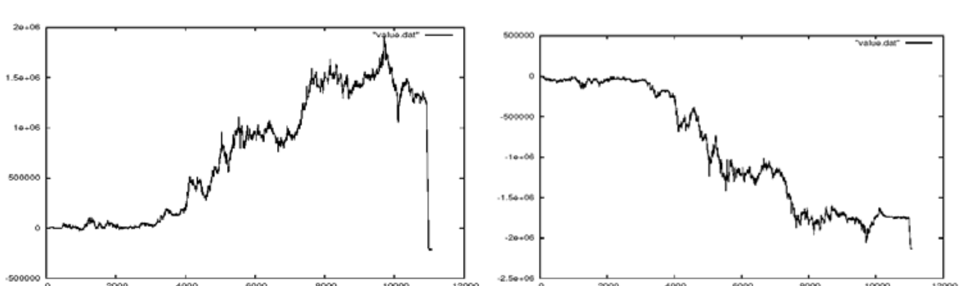


Figure [2] The withdrawal effect: The final agent (left) which withdraws past orders time to time performs much better than the one that doesn't do the same (right).

The observation that orders that increase the liquidity in the PXS server are rewarded in the form of a rebate leads to a small change in strategy. Whenever the contemporary decision is to sell, all previous orders are withdrawn, except for the previous sell orders at prices above the present island price. A similar strategy is incorporated for the contemporary decisions to buy.

4.3 The Volume Parameter

A similar reasoning could have been applied in tuning the volume parameter, in making it a function of the local demand or supply. But on second thoughts, one would notice a fundamental difference in the two parameters. Whereas the price decides whether the placed order would get through and be cleared, the volume is just a multiplying factor of the profit or loss made in a transaction. As a matter of passing, it is useful to mention that the volume has an upper bound of the total volume available for trade.

The contrast is being considered between the following two strategies:

The volume being a function of the differences in island and simulator prices; viz:

eg. buyorder (BUY, *volume*, price) and
eg. sellorder (SELL, *volume*, price)
with *volume* = const * abs(IslandPrice - SimulatorPrice)

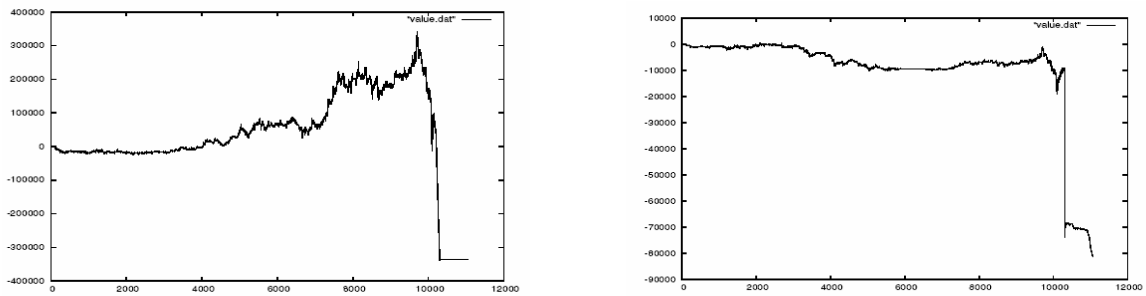
The trader in the competition, with volume set to infinity. Infinity translates to a very high number like 10,000 for practical purposes of the competition.

eg. buyorder (BUY, *volume*, price) and
eg. sellorder (SELL, *volume*, price)
with *volume* = *infinity*

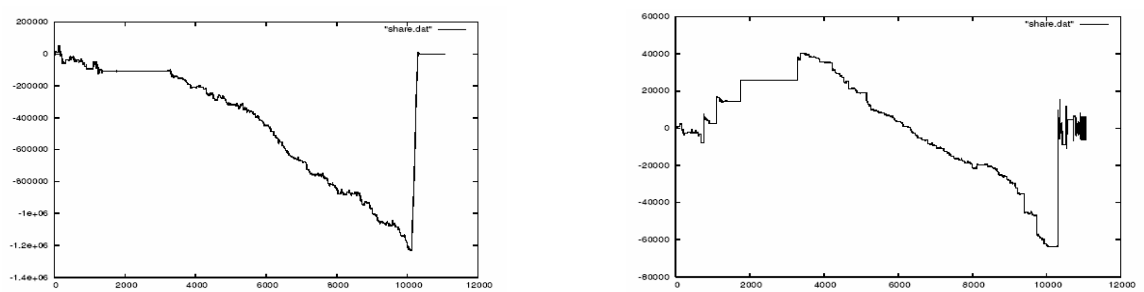
The following are the graphs comparing an agent that decides the volume based on the difference in the simulator and island prices, and that with a constant high volume. It can be seen that the latter outperforms the former.

It is interesting to note that the competitiveness of the strategy, and hence of the orders of an agent is not reduced by changing the volume parameter. Hence, the agents are theoretically expected to have performances on the same lines. The only change expected would be in the magnitude of profits/losses and volumes of shares traded. Indeed, if we take a look at the graphs of the value [3.1] and volumes [3.2] of shares traded, we notice that they look almost the same. That is to say, their behavior is the same. Just the magnitude of the shares traded is different in the two cases.

Figures[3] The Volume effect: The performance of the final agent that attempts to push the maximum possible volume through the order books (left) compared to one with the volume parameter tuned to follow the price difference between island and simulator(right).



Figure[3.1]: Value of shares traded. The profits and losses are magnified to maximum extents.



Figure[3.2]: Volume of shares traded. 1.2 million (constant volume) in contrast to 60 thousand (adapted volume)

4.4 Cumulative results

The following is the compilation of all the results in a tabular format. The itemized values show an average over 10 days of trading.

"Knob" Name	Final Agent		Intermediate Agent	
	Value *(in Dollars)*	**Volume** *(magnitude)*	**Value** *(in Dollars)*	**Volume** *(magnitude)*
Price Parameter	$5.5 * 10^7$	$3.1 * 10^7$	$-4.1 * 10^7$	$8.1 * 10^6$
Withdrawal Effect	$1.8 * 10^6$	$4.2 * 10^6$	$-3.3 * 10^6$	$2.2 * 10^6$
Volume Parameter	$1.1 * 10^6$	$4.7 * 10^7$	-77,978	775,058

Table [1]: Experimental results. Values averaged on training over 10 different days.

5. Related Work

The reverse strategy discussed in Two Stock-Trading Agents: Market Making and Technical Analysis [3] is a strategy that is very similar to my idea. My strategy is a generalization of the same. The reverse strategy plays against the expected market mood. For example, the strategy buys when the market is on the fall. My agent would do the same, not noticing the actual price of the stock in the market, but by noticing that all traders seem to be selling. Since the market mood is to sell, it would play against and buy. A point to note is that my strategy would take factors other than the market order-book statistics also into account, since it bases its decision on the actual market sentiment and not the driving force behind it.

6. Discussion and Future work

Fine tuning the parameters is a crucial part of making profits. I would like to incorporate external factors in tuning the parameters. If possible, I would like to make these parameters changeable in real time. These can be changed in accordance to some external factors. For example, consider a knob for controlling the volumes for buying and selling. The buying and selling knobs could be separate and independent. Further consider the external effect in the form of news stories. An online analysis of a latest news story could predict a downfall of the market. In this case, it could be useful to lower the buy-volume knob and increase the sell-volume knob.

I hope to use news and other external factors in real time tuning of volume, price and other parameters.

7. Acknowledgements

I thank Dr. Peter Stone for all his guidance throughout the project. Discussions in his graduate class, Agent Based Ecommerce have provided very useful inputs in forming the strategy. I also thank the PLAT group for providing me technical support for the simulator.

References

[1] The Penn-Lehman Automated Trading Project:
http://www.cis.upenn.edu/~mkearns/projects/pat.html
[2] Island: http://www.island.com
[3] Y. Feng, R. Yu, and P. Stone, AAMAS-03 Workshop: Two Stock-Trading Agents: Market Making and Technical Analysis
[4] Bloomberg: http://www.bloomberg.com
[5] R. Yu and P. Stone, The Fifth International Conference on Electronic Commerce (ICEC-03): Performance Analysis of a Counter-intuitive Automated Stock-Trading Strategy
[6] Yahoo Finance: http://finance.yahoo.com