Automated PLAT Trading Agent Using Order Imbalance in Volume

Harish K Subramanian

Abstract

Volume of trades and order book volume imbalances have long been established as important criteria in evaluating portfolios and long term investment strategies. The hypothesis being evaluated in this project is that it is an essential component of intraday trading strategies – important enough to be effectively used exclusively as an indicator of the market behavior. The design of a trading strategy based on order imbalance in volume with a genetic algorithm used to tune order volume as a parameter is discussed. The performance of this agent in various environments and some preliminary data from a live competition with other agents is studied.

Introduction and Early Agent Design

A multitude of day trading strategies from 'resistance and support' to the 'market making with volume control' [3] strategy discuss volume as a parameter, in the former case, to aid the decision process and in the latter, as a control mechanism. Most studies however, have considered this a secondary factor and hence, literature on studies of its exclusive effect on intraday trading is scarce. Volume effects have long been studied as a factor in long term investment decisions and portfolio management [4], [5]. In this project, the initial hypothesis was that an agent might be expected to decide trading activity based solely on volume. However, this hypothesis was quickly revised, since any decision to trade stocks brings with it two questions – when to trade and how much to trade. Of course, if the 'when' is answered by a volume measure (order imbalance) [6], then the 'how much' needs to be a measure of price – and this led to the conclusion that this decision is dependent on price and hence price can be adapted as a factor in tuning the order volume – thus answering the 'how much' question. For this, we use a simple genetic algorithm. This pushes one of the parameters to be decided into an independent search space and hence the monitoring of these two parameters can be performed simultaneously. Early work in parameter tuning for trading agents paved the way for this development [7], [8].

Before it was decided that the decision was to be made on volume alone, a number of vistas were explored, most prominently, the effect of 'human' factors such as loss aversion [9] (a concept stemming from Behavioral Economics). Specifically, it was hypothesized that if loss aversion could be modeled into the utility function of an agent, then the agent would decide, at each point, what his utility would be from the trade, and decide based on this. Of course, a loss aversion term would mean this decision would be one of minimizing losses rather than increasing profit, a concept consistent with loss aversion, but redundant in the day trading scenario. A major hurdle was modeling the utility function adequately. And if that was a big problem, a more fundamental flaw in this reasoning came to the fore. While the behavioral effects are pronounced in investment strategies and portfolio balancing (where endowment effect plays a huge factor too), it is difficult to see how this affects day traders. The day trading market is made up of agents (human or automated) playing the margins and this runs counter to the argument that a trader would be loss averse based on the fact that he does not want to lose something that is his already (for real or on paper). However, if the trader obtains the stocks under the assumption that he is going to sell it (maybe in the very next trading cycle!), then this argument seems flawed. At this point, with the deadline for agent development fast approaching, this line of reasoning was abandoned and study on the volume based agent began.

The PLAT Environment and considerations for design of strategy

There were some key changes to the rules for development of an agent in the PLAT [1] environment this year over the last. The PLAT [2] environment focuses on day trading and hence, quite reasonably (as a platform for testing new strategies, extreme exaggeration for the sake of maintaining strategy design integrity is acceptable) has a strong penalty for not divesting one's holdings at the end of the day. A deliberate attempt to force agents to avoid a long or short position was in place.

The performance evaluation this year uses a single criterion for the competition, which is the *Sharpe ratio* of an agent's multiple day profit and loss. For 'n' days of a competition, the profit/loss (p) is calculated each day as:

Profit/Loss = Cash - Penalty + Rebate - Fees.

Here, the cash and penalty are calculated at the end of the trading day, and the penalty is such that a short position would imply you buy the number of shares the agent is short - at *twice* the closing price, and a long position at the end just means you give up all the stock and these shares will be valued at *zero*.

To encourage adding liquidity to the order books, every order that was already in the books is given a *rebate* of $0.002 per share when matched and any order that is matched as soon as it comes in means the agent has to pay a *transaction fee* of $0.003 per share.

The Sharpe ratio for an agent is thus calculated as

{*Avg (p0, p1... pn) / StDev (p0, p1... pn)*}.

Also, there are no limits on how many shares an agent can buy or sell in a day, so this allows for greater exploration of strategies, while the focus on liquidity makes sure that an unreasonable long/short position is avoided.

The rules stated above introduced a few considerations for the design of a strategy, one being an incentive to place orders below the first in the order book. Also, care must be taken while wandering off too long or short in share position as this makes it difficult to liquidate without taking a cut.

The Volume Order Imbalance Difference (VOID) Strategy

The logic behind the basic strategy is fairly intuitive. The order books have buy (bid) and sell (ask) tables. The difference between the total volume of all unmatched orders in buy and sell tables is the order imbalance at any given time. This imbalance is compared with the corresponding imbalance at the previous time instant when data from the books is available. If the imbalance between the buy and sell tables increases, then it is an indicator that there is excess demand over supply and the price will increase. Correspondingly, if the (buy – sell) imbalance decreases, it points to the strength of the sell order book and excess supply over demand. This is bound to drive the price down.

The decision to buy or sell brings with it the issues of *how much* to trade and *at what price*. The volume of trade at each decision point (when the buy/sell order is to be placed) is decided by a genetic algorithm that tunes the desired volume of trade based on previous runs (*weighted* towards the most recent trading days to bias it towards recent trends). The problem of price at which the order is to be placed was resolved so as to maximize the rebate and minimize the fees while still staying relatively close to the top of the order book so the chances of the trade being made are high.

Basic Approach

The basic strategy involves obtaining the order book data for each update (in time, say t1) and getting the corresponding volume of unmatched shares on the buy and sell sides of the book. The difference of these volumes is the buy-sell difference. Then, this same difference is obtained for the next time update (say time t2). The decision rule is stated as

If $(buy - sell)_{t2} > (buy - sell)_{t1}$
  buyOrder(buyprice, ordervolume);
Else If $(buy - sell)_{t1} > (buy - sell)_{t2}$
  sellOrder(sellprice, ordervolume);
Else do nothing

Order Volume

The volume to be ordered at each decision point was tuned using a genetic algorithm (GA), which used, as a fitness function, a measure of the *net profit* made through the day. So, in essence, the volume is tuned for a good profit/loss measure through the day (and correspondingly a good Sharpe ratio), and is updated with every new day of training that it evaluates.

The GA operates by selecting a set of candidate volumes (chosen randomly initially) that is tried one by one as the volume for a given (past) trading day. The day's trading logs are used to obtain the points where buy and sell orders were placed and the corresponding market price for that timestamp. This gives us a series of points in time where orders (buy or sell) were placed as well as the corresponding bid/ask price that was used. When this price is multiplied with the order volume at the time, we get a *cash* measure at the time (positive for sell and negative for buy orders). If the c*ash* is added up through the trading day, we obtain the *net cash* for the day. Additionally, excess volume of shares left over at the end is penalized using a penalty function which is the difference of the total shares bought and total shares sold through the day times twice the price. This ensures a penalty function similar to the penalty while calculating the performance of the agent.

So the *fitness function* used is:

$$\sum[\{(sellprice * ordervolume) - (buyprice * ordervolume)\}(t)] - [2 * price * \left|\sum vol1 - \sum vol2\right|]$$

where *vol1* and *vol2* are the buy and sell volumes respectively.
This fitness function is denoted as *net profit*. Our aim is to maximize this profit.

The fitness function is evaluated for each candidate volume in the population. The winning volume is chosen as the winner of the generation and is chosen to proceed to the next generation. For this, we assume this volume is the mean of a Gaussian distribution and generate a new population of volumes based on this distribution. This ensures that we don't have to start searching for volumes from an altogether new point. Since there is strong correlation between the trading patterns of consecutive days, it is sensible to start searching from the last trading day. Also, this ensures that the latest trading days are *weighted* more heavily in the tuning process, as intuitively it should be.

The original intention was to implement this algorithm online, so it would periodically update the agent. However, this objective was not achieved and the Dec competition was run with the volume tuned overnight with the GA and the agent updated for the next day.

Variable Volumes for Fitness function

Another factor considered for the fitness function was different volumes for buy and sell order volumes. This would yield a fitness function of the form:

$$\sum[\{(sellprice * vol1) - (buyprice * vol2)\}(t)] - [2 * price * \left|\sum vol1 - \sum vol2\right|]$$

Lastly, we used a fitness function evaluated over the day when the day was broken up into segments of different trading patterns. The fitness function in this case becomes:

$$\sum\{\sum[\{(sellprice * vol1) - (buyprice * vol2)\}(t)] - [2 * price * \left|\sum vol1 - \sum vol2\right|]\}$$

Here, *net profit* is computed individually for each section and summed up. Empirically, it has been noticed that most days, the day begins with a trend upwards that drops off after the first hour or so. And towards the end, frantic activity in the market means that it becomes more unpredictable. To allow the GA to tune for these segments separately, we broke up the day into multiple segments and allowed the GA to run for these segments separately thereby allowing a different order volume for each segment. Indeed, it was proven by the GA that the rate of trade (order volume) initially was high and towards the end was low – consistent with the empirical observations.

Order Price

The buy or sell order price was chosen such that the order went into the book *not* above the highest order on the book, thereby avoiding the transaction fees and maximizing the rebate. Hence the general rule used was
buyprice = lastprice – 0.01
sellprice = lastprice + 0.01

Other values including *orderbook prices* were tried. Experimentally, it was observed that mostly, this was the same as the metric used currently, and on the occasion that it was not, this strategy would not want to place an order in the other direction anyway. Another option tried was using the *currentprice*. However, immediate matching of orders causes a failure to capitalize on rebates. Moreover, the decision rule *predicts* which way the price will move at a later time. So, the price is expected to trend the same way that it has been doing before moving the predicted way. In other words, the volume swing *precedes* a price swing and so we can expect a small movement in the direction opposite the *predicted movement* which we capitalize on by placing the order below the first position in the order book. Of course, this being an empirical observation, it is difficult to guarantee such an effect or even say with any certainty that this is the behavior. However, experiments on these various order prices supported the decision of going lower than the top of the order book in terms of order price.

Order Volume Tuning

The GA (in Matlab) tuned the volume parameter to achieve a high fitness function. Table 1 shows the number of generations needed to converge on a reasonable fitness. Table 2 displays the same data for the segmented case (when the day was split into three zones of activity). Of course, it's difficult to say if fitness achieved is a local optimum (which is prone to occurring because of the *elite* nature of producing a new generation). If one of the generations reaches a local optimum, then since the next generation is produced from this result, it is unlikely it will reach a search space that gets it out of this area.

The variability in the number of generations it took to reach a reasonable fitness was large. In most of the cases, however, running an agent with a reasonable number (30 or so) of generations seemed to suffice. For the data in tables 1 and 2, the GA was run for about 50 generations. Figure 1 shows the best and worst case plots in term of convergence to an optimal volume. The GA used to tune the volumes for the live competition days runs for 100 generations.

A distinct advantage was observed in the use of the segmented days and different volume parameters for the segments. The second and third segments (covering most of the day) had their volume controlled as a fraction of the current share. While having a fixed volume parameter in the beginning helped in reaching a fairly large position, this position could then determine what the trading activity should be so as to enable liquidity at the end.

Day	No. of generations to converge	Average no. of generations to converge
1	4	4
2	14	7
3	9	5
4	22	9

Table 1: Number of generations taken to converge to optimal volume measure for fixed segmented volume parameters respectively.

Day	End Share position with fixed volume	End Share position with segmented volume	Total shares traded with fixed volume parameter	Total shares traded with segmented volume parameter
1	-123086	812	294988	598764
2	-15238	-234	441093	809862
3	-21234	-1198	360043	773193
4	-4771	211	345093	609423

Table 2: Comparison of performance of fixed and segmented volume parameter agents

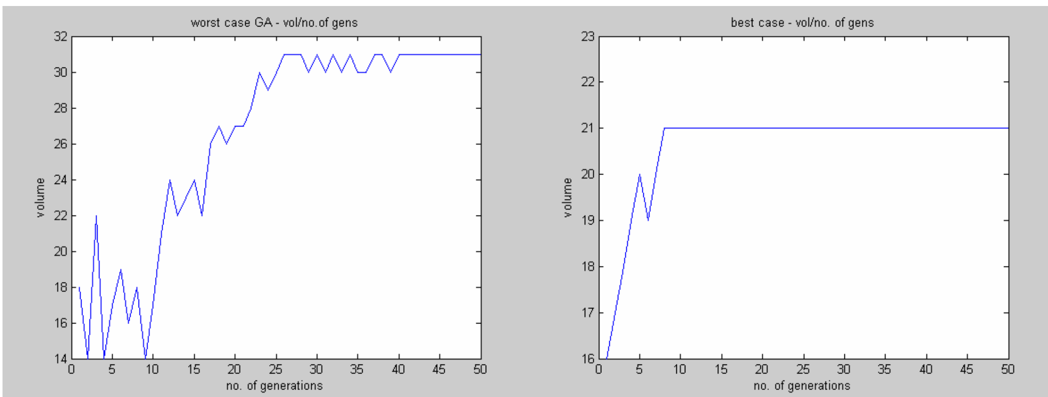


Figure 1: Convergence of the optimal volume (a) Worst case taking about 28 generations (b) Best case taking about 4 generations

This helped achieve a much better result in terms of liquidity and number of total shares traded.

Tests with a SOBI agent, and two versions of the fixed volume agent were run and results compared with the SOBI run with two segmented volume traders. The most effective test, however, was when the fixed volume trader and segmented volume trader went up head to head and against the SOBI. Table 2 gives the results of this run for 4 different days.

An important consideration in the design of the agent was whether to add a deliberate unwinding component to the agent or not. In the end, it was decided that *no deliberate unwinding strategy* will be included because that would make it impossible to gauge ho well the strategy was doing by itself (in terms of volume). As of now, a penalty function is included into the fitness function in the tuning process. This is supposed to tune the volume so that holdings at the end are minimal at the end. In order to study this effect, this component was deliberately excluded. However, in the live competition run by University of Pennsylvania, since no tuning will be effected, a deliberate "dumping" of all holdings may be included.

High Volume of Trades

Very high volume of trade in the market coupled by a period of unidirectional price increase/decrease indicates continued unidirectional trade. This effect can be explained by the occurrence of product releases, earnings announcements, bullish and bearish announcements or other “private” information that an agent might use to trade one way or another. This should be an overriding factor in the agent decision process, and thus prevent the agent from going very long or very short without any hope for recovery. In other words, if the market looks like it will close at a much different price from the starting price without much fluctuation, the basic strategy would go to an extreme long or short position.

Two options were explored in the implementation of a consequent *guard* mechanism. The first was to slow the rate of trading down on detection of such behavior and “lay low” and check again after a while.

The strategy employed for this was:

```
check totalvolume
if (totalvolume is very high)
    if (price increases for three time steps)
            buyvolume = buyvolume/4 ;
            sellvolume = sellvolume;
    else if (price decreases for three time steps)
            sellvolume = sellvolume/4 ;
            buyvolume = buyvolume;
Else do basic strategy
```

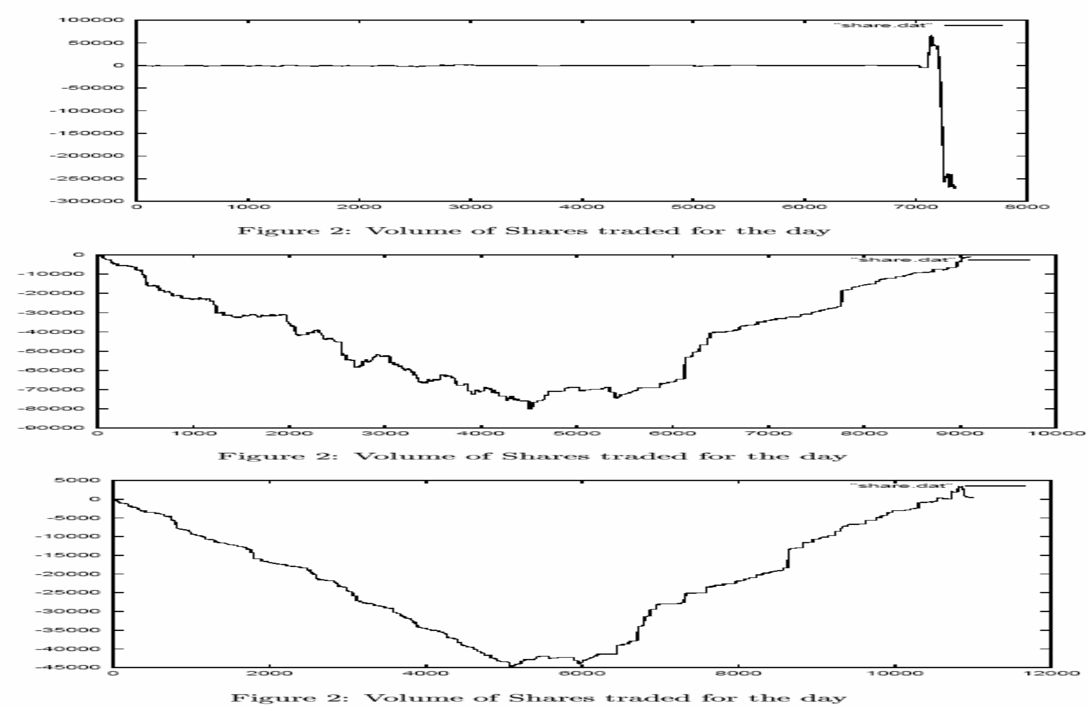


Figure 2: Effect of *guard* mechanism with (a) low threshold (b) high threshold and comparison to (c) no *guard* (basic strategy).

The second was to stop trading if large volume was coupled with unidirectional movement and check a short while later to verify the trend or proceed (false alarm). In both cases, if a unidirectional trend was spotted, a "cut the losses" strategy was employed, i.e., sell if price goes up and buy if price goes down (with the order volumes weighted to try and get to zero holdings). The "do nothing in presence of high volume and unidirectional trade" scheme was implemented as:

check totalvolume
if (totalvolume is very high)
    if (price increases for three time steps)
        buyvolume = sellvolume = 0;(for 3 time steps)
    else if (price decreases for three time steps)
        buyvolume = sellvolume = 0;(for 3 time steps)
Else do basic strategy

Figure 2 shows the volume of trades through the day – for a case when the threshold volume for the *guard* mechanism was quite low, and the strategy does largely nothing till the end (when a reduction in volume causes a release from this condition). In Figure 2, we see that the trading is minimally affected if at all by the *guard* mechanism when the threshold is high – this is reflected in its similarity to the plot in figure , which shows the run without the *guard* mechanism. In both of the above *guard* mechanisms, the difficulty was in ascertaining what "high" volume was exactly. Too low a value caused the agent to suspend trading for no apparent reason. Too high a value did nothing to the process. Hence, it was hypothesized that it is a better scheme to employ for long term trading (where a constant increase could be over sustained periods of time after averaging out small fluctuations) than for intraday trades.

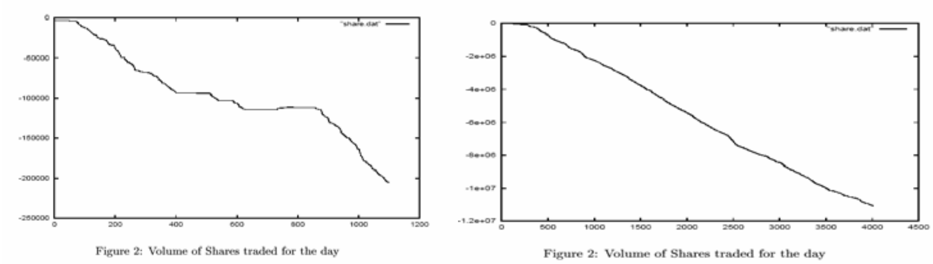
Performance Analysis on Historical Data

Most of the development so far has been done using historical data. In this section, we'll walk through the most prominent results.

Round 1: The first round of tests was done with the most basic agent with no algorithm for tuning of order volume in place. The agent was run against the basic SOBI agent over a series of days in historical data. Most of the data that was churned out wasn't of significant value in terms of reporting, as they were primarily development data that was constantly reworked. At the end, a head-to-head competition was run between SOBI and the basic agent in historical mode. Table 3 shows the results of this round of runs in terms of share position (values are approximate – they just give an idea of the magnitude). What seems evident is that the basic strategy just reacts to SOBI and since there were no other agents, it does exactly the same as SOBI.

Day	Share position (SOBI)	Share position (VOID)
1	- 800000	-10 mill
2	-640000	- 6 mill
3	-725000	-8 mill

Table 3: Round 1 results between SOBI and basic agent without volume control.



(a) (b)

Figure 3: Output of Round 1 in terms of share position (a) SOBI (b) basic agent

Round 2: The second round of tests was done with the basic agent and some volume control with the algorithm for tuning of order volume (without penalty for long/short positions) in place. Again, the agent was run against the basic SOBI agent over a series of days in historical data. There wasn't a qualitative change in behavior since the 'unwinding' aspect was still not in place. Hence, the graphs, while more matched, looked similar to the ones in Figure.

Round 3: The need for a more realistic market emerged by this time, and testing the algorithm was pointless without some kind of realistic behavior on the part of the agents. Also, at this point, feeding this data into a genetic algorithm would be detrimental to the tuning process. So, in this round, multiple agents were run with different starting behaviors, i.e. in the first few hours, these agents behaved very differently because some basic 'reverse strategy' and 'direct strategy' elements were included in the agents. This was done a few times and the agent's previous day output file was used to extract the required information and these were input to the genetic algorithm to obtain revised estimates of the optimal volume. The runtime of the GA was extremely long and only a limited number of these runs could be made, and it was decided that the agent should be run with other agents to have any chance of tuning the parameters to a reasonable level.

The agent was now run for 3 consecutive days in a competition with SOBI, reverse strategy, a direct strategy and a previous hand-tuned basic strategy. The three days were deliberately chosen to be of different nature, i.e. one closed above opening price, one finished below and one finished about the same. Of course, the limited pool to choose from made it difficult to choose a set of days that were dramatically biased one way or another, but these samples proved reasonable for the test. The results of these runs are shown in table 4.

Table 4 : Results of agent in terms of cash and share positions on three different days

Day	Cash position	Share Position
1 (Down)	-734	-2
2 (Up)	242011	-33610
3 (Mean)	-1242	684

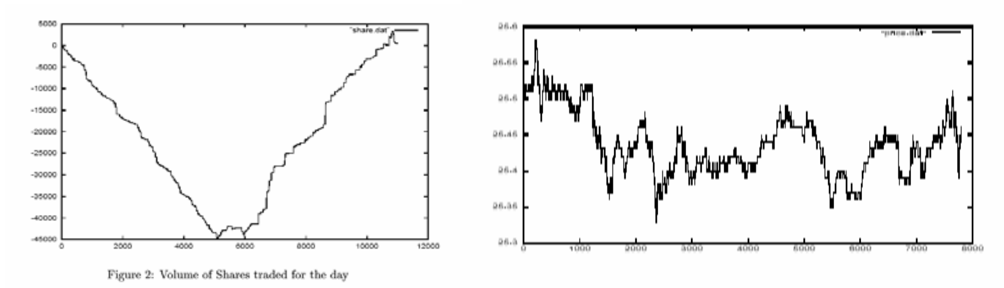


Fig 4: (a) Agent share holdings (b) MSFT 'mean' day

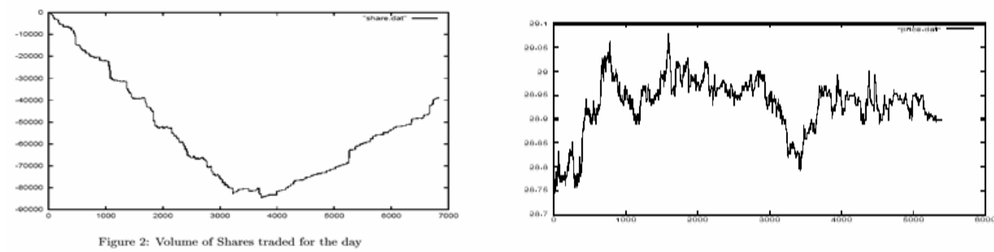


Fig 4: (c) Agent share holdings (d) MSFT increasing day

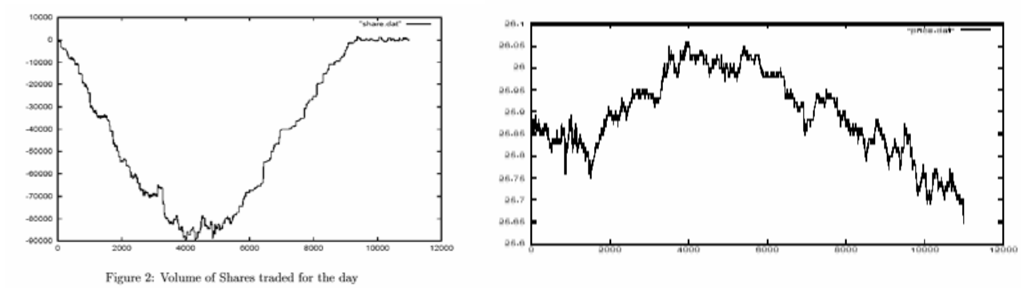


Fig 4: (e) Agent share holdings (f) MSFT reducing Day

Since the GA had far more days when the price decreased or remained constant it seemed much better tuned for these occurrences and much less for the case when the price increased. The training needed to be a lot more intense to reach consistent results even in this limited, repeated, predictable environment.

Round 4: The final test on historical mode was a comparison between the GA tuned strategy and the basic strategy with manual tuning. For this, various setups were considered:
Set 1 - Basic VOID, GA tuned VOID and 2 dummy
Set 2 - Basic VOID and 2 dummy
Set 3 - GA tuned VOID and 2 dummy
The last 2 cases were to make a comparison without the effect of the other agent in question (basic or GA tuned as the case may be).This comparison was easy. The GA

tuned agent did consistently better, and an interesting observation was that it traded larger volumes overall consistently over the same times and thus made more cash and came much closer to unwinding its position consistently. A comparison of GA tuned and basic VOID for these 3 sets can be made from table 5.

Setup	GAVOID Total Volume	VOID Total Volume	GAVOID Cash	VOID Cash	GAVOID Share position	VOID Share position
Set 1	148552	88237	-2457	-1234600	246	11043
Set 2		64991		160880		- 7121
Set 3	109002		-234		21	

Table 5: Performance comparison of VOID agent with and without GA tuning of volume

Performance Analysis of the agent in Live PLAT Competition

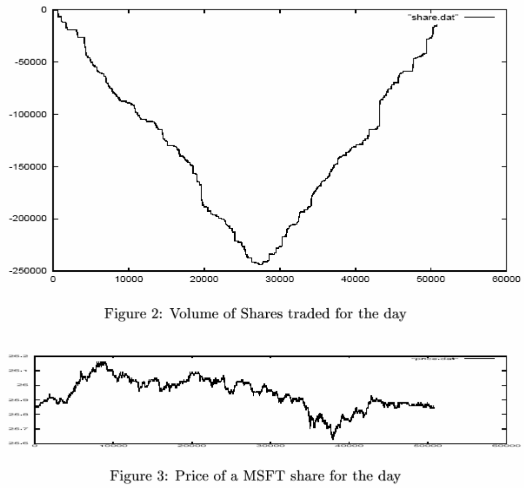
To enable a competitive run of the agents in an environment with unknown, unpredictable agents is essential to really test the agents' performance. A competition run in live mode between all five of the class agents is currently under way, and from the three days' results so far, there is already much information to be had. The competition dates are Dec [1-5].

While evaluating the VOID agent in such a competition, it must be kept in mind that the agent is *expected* to behave well overall, i.e. it averages a reasonable profit. However, due to its propensity *to follow trends of previous days* (weighted towards the most recent), the performance may vary from day to day. Of course, what will be most interesting is to see if the agent averages a profit even when not tuned further (in the University of Pennsylvania competition).

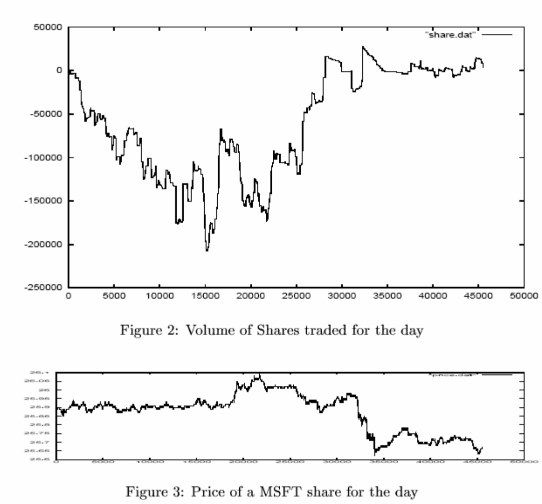
The share plots and MSFT performance for days 1, 2 and 3 are shown in Figures 5(a), (b), and (c) respectively. The cash, share position and net cash after penalties, rebates and fees are shown in table. The day 1 results here do not reflect the true behavior of the agent, since the code had a bug . Moreover, one agent (a high volume trader at that) did not connect and hence the results may be vastly different from the other days. This can be verified when further tests are run on the same day in historical mode. This result and that of days 4 and 5 are included in the *Appendix*.

From these live results, we can make a few preliminary observations that are quite interesting.

- The agent seems to be doing better with passing days. This could be because of the GA (it is run in the night after a trading day) that is tuning parameters better or just a random occurrence due to the nature of the trading days. It is hard to verify this at this early stage. However, if the result is due to the GA, then it would be encouraging and lead to implementation of a GA so it can learn online, and have it in place by the PLAT competition in spring.



(a)



(b)

Figure 5: Share position and MSFT progress n live mode on (a) Day 1 (b) Day 2

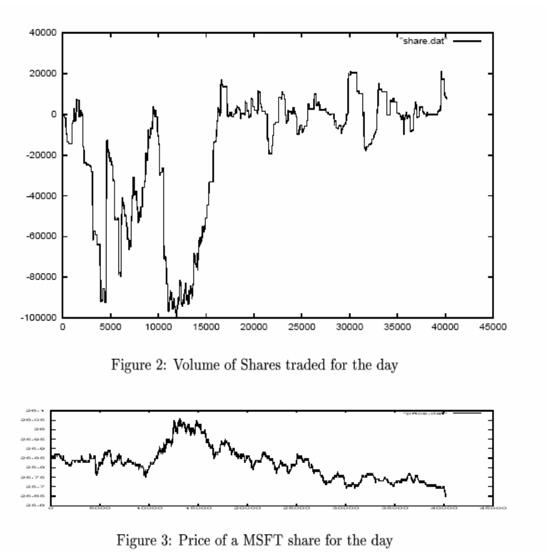


Figure 5: (c) Share position and MSFT progress in live mode on Day 3

Day	1	2	3
Cash	414287.2	-12174	-240.32
Share Position	-14398	851	2323
Rebate	1417.33	3759.48	2978.96
Fees	3.31	22.89	42.096
Net Profit	-327921	-8431.37	2696.54

Table 6: Cash, share and net profit values for the 3 live days

- The agent's total volume traded depends a lot on overall trading volume, i.e. as the total volume traded by all agents in PXS increases, the agent's total volume traded increases. While this is obvious from the nature of the strategy, this means that for any given trading day (except unilaterally increasing), the agent will gain more profits from high volumes of trading. It would be interesting to test this hypothesis. An interesting observation would be to see how it performs on a few days when the parameters are not tuned further.
- As expected, the *rebates* make up most of the transaction proceedings. This means that the strategy of placing orders low on the order books pays off and on some days, like on Dec 3 (day 3), it even brings the agent to profitability. The

rebates caused the agent to end positive on the day despite ending negative in cash holdings.

- Presumably, the agent would not do as well if the GA were not present in the mix to tune the parameters. However, a good test for this would be to run the agent in the live PLAT competition and see if this is indeed true.
- A fairly major flaw in reasoning was fixed at the end of trading on Day 1 which was fixed. Also a minor flaw was fixed on Day 2. So, it is hard to say without a controlled experiment what effected the change in performance. Day 1 is scheduled to be repeated in historical mode, and this might be a useful run to study.
- The performance of agent varies between historical mode and live mode. This may be because the time it takes to run in historical mode is much smaller and hence there is a good chance that the number of orders placed and timing of these orders are very different. Overall, it was found that the trading volume increases in live mode. Another implication is that since the testing is done in historical mode, the only real tuning that helps the GA is the data from live days – and so the performance should improve with further tuning on live days – further cause to develop the online algorithm.

Discussion and Future Work

The VOID strategy is extremely simple and hence, may be bound on how well it can hope to achieve profits. While it tracks volume and places orders on every cycle, it is still largely a trend monitoring mechanism (even if it is a short term trend – like in the intraday case here), and so averages out some of the profits (on the up side, it does the same with a corresponding loss). Overall, volume seems to be a very good indicator of trends over time, but not as good a predictor as to take advantage of intraday fluctuations. For example, if the price was centered around zero most of the time, this would normally suggest a very low order book imbalance and this agent would make very little money. The best result is when there are large fluctuations *and* the price moves above and below the starting price. As with all agents, it makes more money in some conditions than in others. However, it seems to do badly on more days than it does well. This necessitates coupling other factors in with the volume imbalance while evaluating the orders.

The changes to be made to the strategy to make it more profitable seem obvious – the GA needs to be made to learn online and update the agent's order volume periodically. Currently, the data is fed each time for a day's run to a Matlab script and the corresponding output is fed into the strategy file.

Additionally, a more consistent performance measure needs to be coupled with the existing strategy to make it more reliable and less susceptible to "unfavorable" conditions.

The VOID strategy is governed by overall trading volume, and since the overall buy and sell orders for the day are roughly equal for most days, it 'unwinds' automatically. However, this is never enough because it doesn't take into account order withdrawals and residual unmatched orders. This may be the reason why the agent never returns to an absolute zero holding. This is another issue to be addressed.

Overall, the study has mainly been one of the effect of volume imbalance and volume on intraday trading and the feasibility of building an agent based on volume. However, it seems prudent to include other factors such as price variation into the decision making process to improve the predictive powers and help the agent perform better.

Acknowledgements

I would like to thank Dr Peter Stone for his support and feedback through the semester. I would further like to acknowledge Dr Michael Kearns, Berk Kapicioglu and Yi Feng for their continued support, for allowing us to participate in the competitions and the use of the PLAT simulation environment.

References

[1] The Penn Lehman Automated Trading Project 2003 – 04 http://www.cis.upenn.edu/~mkearns/projects/pat.html
[2] Michael Kearns and Luis Ortiz, "The Penn-Lehman Automated Trading Project", *IEEE Intelligent Systems, Nov-Dec 2003.*
[3] Yi Feng, Ronggang Yu, Peter Stone, "Two Stock Trading Agents: Market Making and Technical Analysis", *Dept. of Computer Sciences, UT Austin.*
[4] T. Chordia, B. Swaminathan, "Trading Activity and Expected Stock Returns", *Journal of Financial Economics 59, 3-32, 2000.*
[5] A. Lo and J. Wang, "Trading Volume: definitions, data analysis, and implications of portfolio theory", *Review of Financial Studies, 2000.*
[6] T. Chordia, A. Subrahmanyam, "Order Imbalance and Individual Stock Returns", *Emory University, University of California, Los Angeles, 2002.*
[7] T. Hellstrom, "Looking for predictability in stock returns", *ATAA Journal, 2002.*
[8] T Hellstrom, K Holmstrom, "Parameter tuning in trading algorithms using ASTA", *Computational Finance, Jan 199, MIT Press.*
[9] D Kahneman, J Knetsch, R Thaler, "The Endowment Effect, Loss Aversion and Status Quo Bias", *Journal of Economic Perspectives, Winter 1991.*
[10] Yahoo Finance – http://finance.yahoo.com.