The Penn-Lehman Automated Trading Project

Michael Kearns * **and Luis Ortiz**

Department of Computer and Information Science
University of Pennsylvania
Philadelphia,Pennsylvania 19104

Abstract

We describe the Penn-Lehman Automated Trading project, a broad investigation of algorithms and strategies for automated trading in financial markets that centers around the Penn Exchange Simulator (PXS). PXS performs a stock market simulation that integrates virtual client orders with limit order data from real-world electronic exchanges. We describe our motivations and interests, the design and architecture of PXS, and the results of several competitions we have designed and held among a diverse set of 14 automated trading clients submitted by over 30 participants.

1 Introduction

The Penn-Lehman Automated Trading (PLAT) Project is a broad investigation of algorithms and strategies for automated trading in financial markets. The centerpiece of the project is the *Penn Exchange Simulator* (PXS), a software simulator for automated stock trading that merges automated client limit orders with real-world, real-time order data that is available via modern Electronic Crossing Networks (ECNs; also sometimes referred to as Electronic Communication Networks). In this environment, multi-client trading simulations match client orders both with each other and with ECN orders, thus effectively blending the internal and external markets. PXS automatically computes client profits and losses, volumes traded, simulator and external prices and inside markets, and various other quantities of interest. One major advantage of a simulation incorporating real market limit order books is that it obviates the need for a so-called *fill model* to predict from price information alone when limit orders would be executed; see the discussion in Section 3.1. To our knowledge, PXS is the first simulation platform that employs real-world order books in a detailed order execution process. (All the necessary financial terminology and concepts are described in Section 2.)

There are several underlying motivations for the PLAT Project. From a research perspective, we are among the growing number of artificial intelligence and computer science researchers with an interest in all forms of electronic commerce, computational markets, algorithmic mechanism design and electronic auctions, and related topics. In addition to a burgeoning theoretical literature [1], this line of research also has a growing platform and

*Contact author. Project web site: http://www.cis.upenn.edu/~mkearns/projects/plat.html systems component, perhaps best exemplified to date by the popular and successful Trading Agent Competition (TAC) [6, 5, 4] [1], which has focused primarily on multicommodity auction simulations. Thus, one primary motivation for the PLAT Project is as a contribution to this line of systems and competition work in automated markets. In this regard, a distinguishing characteristic of the project is its investigation of a real and widely studied class of automated markets and strategies. Indeed, Wall Street has many quantitative traders who do for a living what PLAT Project participants do in the safety of the PXS environment. In the same vein, we are also interested in the design of realistic and challenging competitions in automated trading in financial markets, using PXS as the testbed.

We are also active users of PXS, and are interested in using it as a platform for developing novel and principled automated trading strategies (clients). The real-data, real-time nature of PXS makes possible the examination of computationally intensive, high-frequency, high-volume trading strategies (though this last property always presents challenges of estimating market impact). We are particularly interested in the development of strategies that attempt to make predictive use of limit order book data, including via statistical modeling and machine learning. We are also hopeful that over time, the project will generate a library of clients with varying features (trading strategy, volume, frequency, and so on) that can be used to create realistic simulations with known properties.

There is also a major educational component to the PLAT Project. The project currently has over 30 students developing automated trading strategies for PXS, and regular competitions are held among them. Many of the students are in joint programs between Penn's Computer and Information Science department and the Wharton School; several are from external universities. The 14 strategies currently implemented exhibit the wide range of strategies encountered on Wall Street, including variants on classical ideas from technical analysis, market-making strategies, block trading, machine learning approaches, and many others. The scoring criteria for the competitions emphasize not only profit and loss, but a variety of "good trading" practices such as position limits, risk management and position unwinding.

Finally, the PLAT project is an educational and institutional partnership between Penn and the Proprietary Trading Group of Lehman Brothers in New York City, a group of Wall Street professionals who actively design and implement a rich and sophisticated collection of automated trading strategies. Our Lehman colleagues have provided invaluable scientific guidance on the technical design of PXS and the competitions, and have also acted as mentors to the students on the projects.

The summary for the paper is as follows. In Section 2, we provide the necessary background on financial terminology, exchanges, and market microstructure. Section 3 contains a detailed description of PXS, including its overall functionality, architecture, API, and order matching algorithm. Section 4 describes the PLAT project more broadly, including the participants and results from the competitions held to date. Section 5 gives ongoing work on the project and conclusions.

2 Background on Market Microstructure

In order to describe the PLAT Project, as well as PXS, it is first necessary to understand some details of the underlying mechanics of financial markets and exchanges. These computational and transactional details are sometimes broadly referred to as *market microstructure*. Both for the sake of specificity, and because it is most directly related to PXS, we shall focus on the market microstructure of NASDAQ exchanges and stocks, though many of the same elements can be found in other exchanges, including the NYSE.

[1] See also http://auction2.eecs.umich.edu/researchreport.html.

2.1 ECNs and Limit Order Books

Unlike the NYSE, NASDAQ is an entirely electronic exchange: all orders, whether generated by an algorithm or a person, are sent to NASDAQ via an electronic interface and order routing system, and all matches between buyers and sellers are executed by computer. Another distinction between the NYSE and NASDAQ is the degree of distribution: while all transactions in an NYSE stock must eventually be cleared through a single firm or individual (known as the *specialist* for the stock), NASDAQ permits many firms and individuals to provide markets in NASDAQ stocks. It was this environment that led to the advent of *Electronic Crossing Networks* (ECNs), which are firms providing essentially independent and competing markets for NASDAQ stocks. A typical trader of NASDAQ stocks on Wall Street will track prices and activity simultaneously in multiple ECNs, and might break large orders up over several of them, or prefer certain ECNs for certain types of transactions, depending on differing fee structures. Recently there have been some mergers and institutional efforts towards consolidation, but there remain a number of large and independent ECNs, including Island and Archipelago.

A fundamental distinction in stock trading is that between a *limit order* and a *market order*. Suppose we wish to purchase 1000 shares of Microsoft (whose NASDAQ ticker symbol is MSFT) stock. In a limit order, we specify not only the desired volume (1000 shares), but also the desired price. Suppose that MSFT is currently trading at roughly \$24.07 a share (see Figure 1, which shows an actual snapshot of a recent MSFT order book on Island), but we only want the 1000 shares at \$24.04 a share or lower. We can choose to submit a limit order with this specification, and our order will be placed in a queue called the *buy order book*, which is ordered by price, with the highest offered unexecuted buy price at the top (often referred to as the *bid*). If there are multiple limit orders at the same price, they are ordered by time of arrival (with older orders higher in the book). In the example provided by Figure 1, our order would be placed immediately after the extant order for 5,503 shares at \$24.04; though we offer the same price, this order has arrived before ours. Similarly, a sell order book for sell limit orders (for instance, we might want to sell 500 shares of MSFT at \$24.10 or higher) is maintained, this time with the lowest sell price offered (often referred to as the *ask*).

Thus, the order books are sorted from the most competitive limit orders at the top (high buy prices and low sell prices) down to less competitive limit orders. The bid and ask prices (which again, are simply the prices in the limit orders at the top of the buy and sell books, respectively) together are sometime referred to as the *inside market*, and the difference between them as the *spread*. By definition, the order books always consist exclusively of *unexecuted* orders — they are queues of orders hopefully waiting for the price to move in their direction.

How then do orders get executed? Any time a *market* order arrives, it is immediately matched with the most competitive limit orders on the opposing book. Thus, a market order to buy 2000 shares will be matched with enough volume on the sell order book to fill the 2000 shares. For instance, in the example of Figure 1, such an order would be filled by the two limit sell orders for 500 shares at \$24.069, the 500 shares at \$24.07, the 200 shares at \$24.08, and then 300 of the 1981 shares at \$24.09. The remaining 1681 shares of this last limit order would remain as the new top of the sell limit order book. Note that a limit buy (sell, respectively) order with a price much higher (lower, respectively) than the current ask (bid, respectively) is effectively a market order. For this reason, some ECNs (including Island) do not actually offer a separate market order mechanism. But conceptually, there is an important difference between the two types: a limit order is guaranteed price (if executed) but not execution, while a market order is guaranteed execution but not price (since the books might change before the order arrives at the exchange and is executed).

Note that in this setting, every market or limit order arrives atomically and instantaneously

refresh | island home | disclaimer | help

MSFT

GET STOCK

MSFT go

Symbol Search

LAST MATCH		TODAY'S ACTIVITY	
Price	24.0700	Orders	52,983
Time	14:57:07.72	Volume	10,243,212

BUY ORDERS		SELL ORDERS	
SHARES	PRICE	SHARES	PRICE
500	24.0620	500	24.0690
6,000	24.0610	500	24.0690
5,000	24.0600	500	24.0700
100	24.0600	200	24.0800
1,100	24.0550	1,981	24.0900
100	24.0500	412	24.0900
5,000	24.0500	3,000	24.0980
200	24.0500	500	24.1000
3,294	24.0500	100	24.1200
1,000	24.0500	2,800	24.1400
3,000	24.0430	5,000	24.1400
100	24.0400	1,000	24.1400
5,503	24.0400	5,000	24.1500
2,100	24.0300	400	24.1600
2,800	24.0300	1,000	24.1700
(412 more)		(694 more)	

As of 14:57:16.178

Figure 1: Sample Island order books for MSFT.

— there is a strict temporal sequence in which orders arrive, and two orders can never arrive simultaneously. This gives rise to the definition of the *last price* of the exchange, which is simply the last price at which the exchange executed an order. It is this quantity that is usually meant when people casually refer to the (ticker) price of a stock. Note that the last price of a stock in an exchange may change more slowly than the order books, especially in less liquid stocks.

The market microstructure described above has been around since the dawn of financial markets, with some variations in the details (for instance, NYSE specialists have some flexibility in how and when they choose to execute matches between buyers and sellers). What is more recent is the *automation* of this process in markets such as the NASDAQ. The fundamental role played by ECNs (and by PXS) is the computerized maintenance of buy and sell order books in the offered stocks, automated order execution, and various other related functionality (such as the ability to withdraw or change unexecuted orders, to check the status of a previously placed order,and so on).

What is even more recent than the automation of market microstructure is the *publication* of real-time order book data. The publication of such data presents a number of intriguing opportunities [2], as the limit order books can be viewed as an expression of market sentiment, and more prosaically may provide strategic guidance for order placement. Indeed, it is common on Wall Street for traders to examine the limit order books carefully, and place their orders accordingly (for instance, by "stepping in front" of an existing limit order in the book by just a fractional amount).

It is the availability of order book data that also makes possible PXS. We now describe the well-known ECN from which PXS obtains its external market data.

2.2 The Island ECN

Island (www.island.com) is a major ECN for NASDAQ stocks, and accounts for approximately one of every seven NASDAQ trades. They are thus among the most important providers of liquidity in NASDAQ, and like many ECNs, they are a technologically sophisticated company. In addition to accepting orders arriving through a variety of standard Wall Street brokerage order routing systems, they also provide an API (known as the OUCH protocol) for automated order placement and management.

Island also offers data feeds (in what is called the ITCH messaging format) that provide transaction-level detail on practically all Island trading activity. From such data it is possible to reconstruct the state of the Island limit order books in any stock at any point in time, including currently.

OUCH ports to Island's trading servers and ITCH data feeds are of nominal cost. Like all ECNs, Island's primary source of revenue is the per-transaction fees it collects on executed orders, and thus encourages development of automated and human trading systems that interface with Island servers.

The Penn Exchange Simulator can be thought of as an experimental ECN that merges the Island order books with orders placed by virtual trading clients. Rather than using an ITCH feed, PXS uses order book data available in real time on Island's web site, since this simplifies the process of building the books from transaction-level data (though we are contemplating a switch to an ITCH feed in the future). We now describe PXS.

[2] It has also spawned a relatively new literature attempting to come to both a theoretical and empirical understanding of order book behavior [2, 3].

3 The Penn Exchange Simulator

3.1 PXS Functionality Overview

As we have suggested, PXS merges limit order data from two sources: actively connected trading clients, and limit orders from Island. The Island data is obtained from Island's live web-based Order Book Viewer, which shows the top 15 limit orders (price and volume) in the buy and sell order books. (The fact that the Order Book Viewer shows only the top 15 orders, rather than the entire books, creates some technical challenges in the accurate maintenance of the PXS books, which are discussed later.) PXS simulations may be run either in live mode, in which case the Island data is updated in real time from the Island Order Book Viewer approximately every 3 seconds, or in historical mode, which requires that the requested day of Island data has already been archived. While live simulations provide the most realistic merging of the Island and PXS markets, since they are faithful to the timing details of activity on Island, historical simulations have the virtue of being considerably faster, as PXS can process each update of Island data at its own internal processing rate.

Before discussing the details of PXS, we note that every major brokerage on Wall Street of course has multiple platforms and data sources on which to test new trading strategies; these are often referred to as *backtesting* environments. To date, it appears that most or possibly all of the extant backtesting environments employ price information alone, or in some cases price and inside market data. This is almost certainly due to the fact that order book data has been exposed only very recently (and is still not yet available in many major markets). One major advantage of an order-based simulation platform such as that provided by PXS is that it obviates the need for a *fill* or *execution model*. When only price information is available for a stock, and a proposed strategy wishes to place a limit order away from the current price, any simulation must make a decision about whether such an order is executed at a future time based on just the price data. Obviously, if the price never reaches the limit order price, the order would never get filled — but if the price does cross the limit order price, it might or might not have been executed in the real market, depending on the depth of demand in the books at that price. A typical fill model might probabilistically execute the limit order based on historical data and the volume of the order. In contrast, in PXS there is never any need for such models: just as in the real exchanges, a limit order is filled only if it is matched by an opposing order at some point in the book-based simulation. Rather than modeling the depth at different price levels in the books, we have the books themselves. To our knowledge, PXS is the first simulation tool employing real-world order book data in this manner.

PXS is implemented in C on Unix and Linux platforms, and takes the following arguments when it is invoked:

- The four-letter ticker symbol of the Island-traded NASDAQ stock in which to run a simulation (such as MSFT).
- The port number over which PXS will communicate with trading clients.
- For simulations on archived historical Island data, the date for which the simulation should be run. If this argument is omitted, a live simulation using the current day's Island data is executed.
- If an historical simulation is being run, a flag indicating whether the simulation should be run in *asynchronous* (the default) or *synchronous* mode (discussed later).
- For the trading day being simulated (whether live or historical), the start and stop times for the simulation.

As an example, the command

```
pxs -p 9800 -n MSFT -h 04292003093000 -e 160000
```

would start an execution of PXS on Island MSFT order book data from April 29, 2003 at 9:30 AM of that day, and terminating with an Island update close to 4 PM on that day. This execution would then accept connections from trading clients over port 9800. The same command but omitting the "-h" argument would cause the simulation to begin immediately using the current Island MSFT data.

3.2 Client API

Once a PXS simulation is under way, any number of automated trading clients may join at any time by connecting via the designated port. Each connecting client will be assigned a client identifier by PXS.

The PXS API for clients contains a rich set of data structures and functions that permit the placement and withdrawal of orders in the PXS market, and the computation of certain market information. Technically speaking, the functionality discussed below is actually split between PXS itself and a client shell process inside which participants implement their particular trading strategies. In general, it is PXS which computes quantities of global interest to the simulation, while the client shell computes more client-specific quantities. For ease of exposition, however, we will blur this distinction, and simply refer to PXS.

The most basic client API functions are those for order placement and management:

- **buyOrder**(p, v)**:** Places a limit buy order at price p for v shares. Returns an identifying number for the order.
- **sellOrder**(p, v)**:** Places a limit sell order at price p for v shares. Returns an identifying number for the order.
- **withdraw**(o)**:** Withdraws order with identifier o from the PXS order books. Note that this will fail if the order has already been executed.

In addition, there are functions for computing the following agent-specific quantities of interest:

- The current cash and share holdings of the client.
- The current profit and loss of the client, under either PXS or Island last price valuation for share holdings.
- The volume of orders currently outstanding in the PXS buy and sell books for the client.

There are also a number of API functions providing information about the overall PXS market state:

- The last price of the PXS and Island markets, respectively.
- The current time according to PXS. For live simulations, this will always be close to the actual (wallclock) time. For historical simulations, it will be the timestamp of the most recently processed Island update.
- The total volume of shares and number of orders currently in the PXS buy and sell order books.
- The total volume and number of orders already matched by PXS in the simulation so far.
- Various statistics of the PXS order books, including volume-weighted average prices.

More generally, clients are able to receive a copy of the entire PXS order book data structure, which shows the price and volume of every limit order in the book, along with indications of which orders came from Island and which from PXS clients. Thus, PXS provides clients with a level of internal market visibility which matches that offered on real ECNs such as Island. This permits the development of arbitrarily sophisticated use of market microstructure data by PXS clients.

3.3 The PXS Execution Engine

At the heart of PXS is the execution engine, which is responsible for maintaining the PXS order books, for integrating both Island client limit orders into these books, for executing matching orders of PXS clients, and for computing the share position, cash holdings, and profit and loss of PXS clients.

A detailed description of the main loop of the PXS execution engine is given in Figure 2. Below we give an overview and explain some of the more subtle aspects of this execution engine, including our partial observability of Island activity, and liquidity assumptions.

At a high level, the execution engine consists of a number of main steps that are executed repeatedly for the duration of a simulation:

- The updating of the PXS buy and sell order books with any new orders detected on Island.
- The "cleaning" of the PXS books after the Island updates, a step to address our partial observability of the Island books.
- The execution of PXS orders that are matched by unobserved executions on Island between updates.
- The execution of matches between buy and sell orders in the PXS books, the updating of the PXS last price, and the updating of client share positions, cash holdings, and profit and losses.
- The insertion of newly arrived client orders into the PXS books.

Before diving into some of the details of this process, we make a number of important remarks on PXS client cash and share holdings, and valuation. In any PXS simulation, all PXS clients begin with no cash and no shares of the stock being traded. At any time, a PXS client is free to either buy or sell shares of the stock, regardless of their current share and cash holdings. Thus, clients may sell more shares they hold (selling short), or buy shares without cash. Share and cash holdings may thus be either positive or negative. PXS maintains these holdings for each client.

The valuation (profit and loss) of a client's holdings at any time is defined as the sum of its cash balance (positive or negative), and its share position (positive or negative) times the current (last) price of the stock. Thus, an important (and potentially unrealistic) *liquidity* assumption is made — namely, that at any moment, any client could return its share position to zero instantaneously by placing all shares held on the market, and receive the current price for all of them. For large share positions, this is clearly unrealistic, as the immediate attempt to buy or sell large numbers of shares will move the price unfavorably. Furthermore, there is a choice of whether to compute the valuation of share positions using the Island or PXS last price. PXS computes the valuations both ways, but this issue is discussed further in Section 4.2.

We now begin discussion of the execution engine itself.

Initialization.

- Initialize two empty lists of *(price,vol,id)* order triples for the PXS buy and sell books.
- Initialize an empty list of order triples for the already-executed PXS orders.
- Initialize the *PXS Last Price* to be 0.
- Initialize the *Island Last Price* to be 0.
- Initialize the *Island Total Volume* to be 0.

The following steps are executed repeatedly for the duration of the simulation:

1. ***Get Island Update.*** Poll the Island Order Book Viewer for the stock being traded, and download the current snapshot of the top 15 buy and sell limit orders (prices and volumes) in the Island books. Update the Island Last Price, and the Island Total Volume, from the current snapshot.
2. ***Island Update to PXS Books.*** For each order $o =$ *(type,price,vol,id)* in the Island book snapshot:
   (a) Check if the Island order identifier *id* already appears in the PXS books. If so, change the corresponding PXS order volume to *vol* minus any volume already matched for that order. If the resulting volume is non-positive, remove the order from the PXS books.
   (b) Check if the Island order identifier *id* already appears in the list of already-executed PXS orders. If so, ignore the order.
   (c) If there is no record of the Island order in either the PXS books or the PXS already-executed list, place it in the appropriate PXS book (buy or sell, determined by *type*) in the appropriate (price-sorted) place. The Island order identifier *id* also used as the PXS order identifier.
3. ***PXS Book Cleaning.*** For each order $o =$ *(type,price,vol,id)* in the PXS books:
   (a) Check if the order originated on Island. Orders in the PXS books with positive order identifiers ($id > 0$) are from Island, since PXS client orders are assigned negative order identifiers ($id < 0$).
   (b) If the order originated on Island, see if it *should* appear in the latest Island book snapshot. This occurs if the *type* is *buy* and *price* is higher than the last (15th) price in the Island buy book snapshot, or if the *type* is *sell* and *price* is lower than the last (15th) price in the Island sell book snapshot.
   (c) If the order should appear in the Island book snapshot but does not, then it has vanished from Island due to withdrawal or Island execution between snapshots. Remove the order from the PXS books.
4. ***PXS Order Execution from Island Available Volume.***
   (a) Let the *Island Available Volume* be the difference between the Island Total Volume at the current snapshot and the preceding snapshot.
   (b) Consider all *client* orders ($id < 0$) in the PXS buy book whose price is greater than or equal to the Island Last Price, and all *client* orders in the PXS sell book whose price is less than or equal to the Island Last Price. If the total volume of all such PXS orders is less than the Island Available Volume, then consider all these orders to be matched by PXS, remove them from the PXS order books, and add them to the list of already-matched PXS orders. If the total volume of all such PXS orders exceeds the Island Available Volume, then match them in order of their position on the PXS books, up to a total PXS volume that equals the Island Available Volume. Note that this may cause an order on each book to be partially matched.
   (c) If any PXS orders were matched in the preceding step, update the PXS Last Price to be the Island Last Price.
5. ***PXS Order Execution from PXS Books.***
   (a) Process each order in the PXS books in order, and execute any matches between buy and sell orders that cross.
   (b) Move each matched order from the PXS books to the list of PXS already-matched orders.
   (c) Update the PXS Last Price each time a match is executed.
6. ***Process PXS Client Orders.***
   (a) For every newly arrived PXS client order, check if it can be immediately matched with another order already in the PXS books. If so, match it and update the appropriate PXS books and the PXS Last Price accordingly. Otherwise insert it into the correct price-sorted spot in the appropriate book, assigning it the next negative sequential identifier *id*.
7. ***Update PXS Public View and Get New Client Orders.***
   (a) Update the public view of the simulator state that will be sent to the clients (PXS books, Last price info, etc.).
   (b) Get any new orders received from clients since the last iteration.

Figure 2: Details of the PXS Execution Engine.

3.3.1 Island Updates to the PXS Books

The process of updating the PXS order books in response to Island and internal order placement is conceptually simple, and is detailed in Step 1 and 2 of Figure 2. At the beginning of every cycle (Step 1), PXS first loads the latest Island data (either by directly polling the Island Order Book Viewer for the stock for live simulations, or getting the next timestamped file from the archival store for historical simulations). Recall that this data consists of a snapshot showing the top 15 limit orders in the buy and sell books. Since Island orders have their own identifiers, PXS can easily scan its internal books and see which Island orders are already present in the PXS books, and which are new (Step 2).

For those that are present in the PXS books, PXS checks that the Island and PXS volumes for the order match, and if not, updates the PXS volume (Step 2a). Volume changes can occur on Island due to partial matching of an order, or the Island customer changing the order size. For those Island orders not in the PXS books, PXS simply creates a new limit order, identifies its origin as Island, and inserts it into the proper place in the PXS books.

3.3.2 PXS Book Cleaning

A complication arises in this Island update process due to the fact that PXS has only partial observability (the top 15 orders) of the Island books. If, for example, the Island price rises from p, and with it the prices in the limit orders at the top of the Island buy book, we will no longer be able to see in the Island books those orders that were visible to us when Island was at p. This by itself presents no problem; we continue to treat these orders as extant in the PXS books even though they are too far down in the Island books for PXS to see them. However, if the Island price later falls back to p, then we should now be able to again see the orders we loaded near p from Island — unless, of course, they are actually no longer in the Island books (for instance, due to customer withdrawal). PXS thus uses such a return to an earlier price as an opportunity to confirm the continued Island existence of any orders in the PXS books that "should" be visible on Island. If an order that should be visible has disappeared from Island, it is removed from the PXS books. This process is called *cleaning* the PXS books, and is implemented as Step 3.

3.3.3 PXS Order Execution from Island Available Volume

Aside from our ability to view only the top 15 orders on each Island book there is another, more important sense, in which our view of Island activity is handicapped. By definition, the Island books show only a record of those orders which have *not* been executed already. If an incoming Island order matches an order sitting on the Island books, the execution of the pair will cause the book order to *disappear*, and we may see no trace at all of the incoming order. Even more extreme, two orders may arrive and match each other between our Island updates, and we would see no trace in the books of either order. In a highly liquid stock, such "invisible" transactions may constitute a significant fraction of the trading volume. Obviously, to maximize the realism introduced by the Island data, we would like to somehow recapture this invisible liquidity.

Fortunately, there is a reasonable proxy we can construct via the Island last price and total volume figures. With every update to its books, Island also publishes the current last price (which again, indicates the price of the most recent transaction), and the total volume traded in the stock for the day so far. By computing the change in total volume since our last download from Island, we know how many shares have been exchanged in that interval. While we cannot know the prices of each exchange, in our simulation we simply use the current last price as the presumed price of them all. In a highly liquid stock, this is a reasonable assumption, as price moves in only very small increments on short time scales; and in an illiquid stock, it is also a reasonable assumption, since there are fewer transactions and less volume between updates. The process of computing the recently traded volume on Island, and making those shares available in PXS, is implemented as Step 4.

We note that both aspects of partial observabilty on Island (visibility of books to only 15 orders, and invisibility of executed volume) are artifacts of our exploiting Island's web-based Book Viewer, and would be obviated by recoding the simulator to use an ITCH data feed from Island, which provides transaction-level data. However, we note the following points regarding the approach taken here:

- By limiting our access to the Book Viewer data, we *greatly* reduce the amount of data that we must process. Island book sizes in liquid stocks routinely reach lengths the hundreds or longer.
- It is not clear that deeper book access would mean more realistic simulations, as orders deep in the book are often "stale" and have simply been forgotten by traders who know they have no chance of execution, and will be cleared at the end of the trading day.
- Even with a transactional data feed, there are always "invisible" orders and events on Island and other ECNs.

3.3.4 PXS Order Execution from PXS Books

After the preceding steps, the process becomes simpler: we have now incorporated any new Island orders into the PXS books, and have executed whatever PXS volume can be matched by the Island Available Volume. It remains to simply match any orders on the current PXS books that can be crossed, updating the PXS Last Price accordingly; publish for PXS clients an update of the PXS state; and finally, accept another round of PXS client orders. These steps are implemented as Steps 5, 6, and 7.

4 The Project

So far we have given detailed motivation and description of PXS, which is a complex software system for realistic automated trading simulation. As discussed in the Introduction, however, the development of this tool is just one goal of the overall PLAT Project. Inspired by the scientific and sociological success of multi-agent competitions such as the annual Trading Agent Competition, we also seek to use PXS as a testbed and platform for realistic and interesting interactions between groups of independently developed automated stock trading agents. This section describes our efforts so far in this direction.

4.1 Participants

The PLAT Project has been accepting both Penn and external participants since the Fall of 2002. Aside from the project staff team of five (who oversee PXS maintenance and development, the Island data management, and the running of the competitions), there are now a total of approximately 30 people involved in the development of a wide variety of automated PXS trading clients.

Penn is an excellent source of interest for the project, as there are a large number of undergraduates receiving joint degrees from the Computer and Information Science department and the Wharton School. These students must undertake a year-long senior research project, and many have chosen to do so as participants on the PLAT Project developing novel automated trading strategies.

The competitions have also benefited from the participation of three external teams, two from the University of Texas at Austin, and one from Carnegie Mellon University.

4.2 Competitions

Since the inception of the project, a total of three formal competitions have been held. Although PXS has the capability of performing simulations in any stock traded on Island (which includes all of NASDAQ), all of the competitions were executed on Microsoft (MSFT) stock. This decision was made for several reasons:

- The task of maintaining complete and consistent historical Island data (for running historical simulations) requires a fair amount of management and disk space, and focusing on a single stock minimized this burden.
- For pedagogical purposes, it is instructive to consistently test and compete on a single stock, so teams can begin to calibrate their strategies accordingly, and use the historical data for backtesting.
- Island is an extremely liquid exchange for MSFT, and thus permits simulations which ensure that the internal PXS market is strongly influenced by the real external market.

A common issue for all the competitions was risk management. Just as on Wall Street, it is important to encourage teams to develop strategies that intelligently balance their risk and return. In the setting of the PLAT Project, risk generally comes in the form of large share positions (long or short), since such positions are extremely vulnerable to unfavorable changes in the share price. From the perspective of intraday trading (which has been the exclusive focus of the project so far; the competitions "cash out" each client at the end of each day, with no positions held overnight), an ideal strategy would manage to end each day with large *cash* holdings and a zero or small share position.

In order to prevent the results from being dominated by strategies that simply place large bets in the form of excessive share positions, in all of the competitions there was a firm rule that during any trading day, a client's share position must always remain within a window of $\pm 100,000$ shares. Violation of this was grounds for disqualification; while minor infractions were forgiven, in one of the early competitions there was indeed such a dismissal. This limit is a crude and easily verifiable way of ensuring that no client succeeds simply by taking much larger positions than all others. Since (as of this writing) MSFT trades at roughly $25 a share, the share position limit effectively means that clients may never have more than approximately $ 2.5 million of virtual capital at risk. This would be a rather generous leash for a junior trader in a Wall Street brokerage.

Another issue common to all of the competitions was how to value a client's share position at the end of each trading day. When the PXS market closes, each client will generally have a (positive or negative) cash position (the balance of the cash it has spent on the purchase of shares throughout the day, and the cash it has received on sales), and a balance (long or short) of MSFT shares. In order to convert this portfolio to a cash valuation, we must assign a value to the share position. For large share positions, it is unrealistically optimistic to simply take the share position and multiply by the last price of the stock, as the placement of a market order for a large number of shares obviously may eat deep into the limit order books, resulting in progressively less favorable prices. This is the well-known and difficult problem of assessing the *market impact* (effect on prices) of large orders, and is the reason that accurate measurement of strategy performance from historical data is difficult, and also why brokerages almost always break large orders into small increments over time. While the presence of other trading clients in the PXS market can act as an important simulator of market impact (a topic we shall return to), for the purposes of day-end valuation, we do indeed make a somewhat unrealistic "infinite liquidity" assumption, and value every client's share position at the last price. The assumption is somewhat justified by the aforementioned share position limit on clients, and the fact that we are indeed trading a stock (MSFT) with very high Island liquidity. Other valuation methods are possible, however, and may be explored in future competitions.

The three competitions were held in November-December of 2002, February-March of 2003, and April-May of 2003. In each competition, client strategies were divided into pools, both for the sake of diversity of population, and in order to reduce the computational load of each simulation. The competitions had varying formats, with the first two having no restrictions on clients other than the share position limit, and the winners determined strictly by cumulative profitability over the period of the competition. We shall not discuss the results of these early competitions in detail, but one of the primary lessons to emerge from them was that additional rules or scoring criteria were desirable in order to encourage increased realism among clients. This led to the third competition, dubbed the *Platinum Platter Competition* (PPC 2003), which we shall now describe and analyze in some detail.

For PPC 2003, the 14 entrant strategies were divided into two pools (named the Blue and Red pools) of 7 clients each. The division was made somewhat arbitrarily, but also taking account of coarse preliminary experiments to ensure that there was a reasonable amount of client liquidity in each pool.

The competition was held for each of the 10 trading days of the weeks of April 28 and May 5 of 2003. Each day's PXS simulation ran during the market hours of 9:30 AM to 4 PM, the normal trading hours of the NASDAQ exchange. (Although Island conducts after-hours trading, the liquidity tends to be considerably lower than during normal exchange hours.) While daily and overall profitability naturally remained important components in the evaluation of clients, an interesting and considerably richer set of scoring criteria were used to encourage client realism and good trading practices. We now describe these criteria.

4.2.1 PPC 2003 Scoring Criteria

Figure 3 provides the details of the scoring criteria used for PPC 2003. Here we discuss the motivation behind each category.

Daily Profit and Loss. This category is the most straightforward way in which clients are rewarded for generating large positive returns at the conclusion of each trading day. Note, however, there are limits to the benefits: since scoring is strictly by rank, and profit does not carry over to subsequent days (in this category), individual days of great profitability (or loss) have limited overall impact.

Overall Consistency Profit and Loss. Here we reward profitable cumulative performance of *any* magnitude — provided there is some minimal evidence for consistency as well. Independent of the formal scoring criteria, below we will also examine some standard statistical measures of consistency.

Daily Intraday Position Reversals. This category attempts to reward strategies that show enough sophistication to (profitably) do both significant buying and selling within a trading day, as opposed to simply going long or short the stock all day. The goal was to encourage teams to think about intraday signals of impending price reversals, and similar issues.

Robustness to Market Variation. Here we reward strategies that can succeed in both bull and bear markets.

Daily Risk Saturation. In the early competitions, we found that several teams overreacted to the 100,000 share position limit, and traded in such small volumes that they were seriously disadvantaged in comparison to clients that also obeyed the limit, but consistently came close to it. We thus introduced this category, designed to encourage competitive trading activity. Two forms of the criterion are necessary, since certain well-known strategy types (such as market-making) will trade very high volumes overall, but in a way that deliberately maintains a share position near zero.

Criteria Emphasizing Profitability.

- **Daily Profit and Loss.** On a daily basis, 3 points awarded to each client whose end-of-day P&L is highest in its seven-client pool; 2 points to the second highest; 1 point to the third highest. Maximum possible award: 30 ($= 3 \times 10$) points.
- **Overall Consistency of Profit and Loss.** A one-time award of 15 points to any client that has positive cumulative P&L over the 10 trading days of the competition, and also ends at most 3 trading days with negative daily P&L. Maximum possible award: 15 points.

Criteria Emphasizing Robustness, with Weak Profitability Prerequisites.

- **Daily Intraday Position Reversals.** On a daily basis, 2 points awarded to any client that finishes with positive P&L for the trading day, and held share positions in excess of 10,000 shares in both the long and short direction at some point during the day. Maximum possible award: 20 ($= 2 \times 10$) points.
- **Robustness to Market Variation.** An award of 5 points to each client that has positive P&L on any pair of trading days in which the share price rose overall (open to close) on one day of the pair, and fell on the other. For each additional such pair, an additional award of 5 points. Maximum possible award: 25 ($= 5 \times 5$) points, if there are exactly 5 up days and 5 down days for the stock during the 10 competition days.

Criteria Emphasizing Good Trading Practices, with No Profitability Prerequisites.

- **Daily Risk Saturation.** On a daily basis, 2 points awarded to each client that achieves a share position in excess of 50,000 shares (long or short) at some point during the trading day, without exceeding the maximum allowed share position of 100,000 shares. Alternatively, these 2 points will be awarded to clients whose total matched volume of shares for the trading day exceeds a fraction 1/14 (which is half of the per-client average of 1/7) of the total matched volume of all clients. Maximum possible award: 20 ($= 2 \times 10$) points.
- **Daily Position Unwinding.** On a daily basis, any client that is awarded the Risk Saturation points for that day may earn an additional 2 points by ending the trading day with a share position of less than 5,000 shares (long or short). Maximum possible award: 20 ($= 2 \times 10$) points.

Figure 3: Client Scoring Criteria for PPC 2003.

Strategy Name	Description	Performance
CBR-SOBI	Case-based reasoning applied to the parameters of the SOBI strategy (see text for SOBI description).	First in the blue pool; statistically significant profitability.
MoneyFlow	Predictive strategy using money flow (price movement times volume traded) as a trend indicator.	Second in the blue pool.
OBMM	Market-maker that positions orders in front of the nth orders on both books.	Third in the blue pool.
CReaTiv	"Capitalization on Real Time Volatility" — SOBI modified by recent volatility	Fourth in the blue pool.
OBCrossover	Exponential Moving Average (EMA) crossover strategy moderated by confirmation of order book quartile VWAPs.	Tied for fifth in the blue pool.
OBBreakout	Breakout strategy applied to trend lines on VWAP of buy and sell books.	Tied for fifth in the blue pool.
RaSTa	"Resistance and Support Trading Agent" — Computes support and resistance levels based on peaks in order book volume.	Seventh in the blue pool.
DAMM-STAT	Mixture of a Dynamically Adjusted Market-Maker which calibrates by recent volatility, and a trend-based predictive strategy.	First in the red pool; stellar position management.
Contrarian	Sells on rising prices, buys on falling prices.	Second in the red pool.
OBSigma	Trades based on relative spreads in the buy and sell books, interpreting small standard deviation as a sign of confidence.	Third in the red pool.
OBVol	Simple predictive strategy using total volumes in buy and sell books.	Fourth in the red pool; highest Sharpe ratio and statistically significant profitability.
RapidMM	Market-maker with rapid revision of quotes based on current inside market.	Fifth in the red pool.
CIA	"Central Intelligent Agent" — Predictive strategy applying Boosting to order book snapshots.	Sixth in the red pool.
SimpleTrend	Simple trend prediction strategy.	Seventh in the red pool; statistically significant negative earnings.

Figure 4: PPC 2003 Strategy Descriptions, grouped by competition pool.

Daily Position Unwinding. This criterion encourages teams to obviate our aforementioned "infinite liquidity" assumption in valuing share positions, by (approximately) cashing out their positions themselves by trading activity in the PXS market. Risk saturation is a prerequisite in order to prevent clients from receiving credit for unwinding a position that was not significant to begin with.

4.2.2 The Strategies

Figure 4 provides a brief description of each of the 14 clients entered in PPC 2003, an indication of whether it competed in the Red or Blue pool, and its overall final pool ranking according to the scoring criteria of Figure 3. We will analyze several aspects of the competition in some detail below, but first make some high-level comments about the diversity of trading strategies.

One theme that emerged was the use of order book information in the trading clients. While PXS itself of course makes use of Island and internal order books to conduct its simulations, it also makes its books available to trading clients in real time, thus permitting strategies that attempt to derive predictive or other value from this information. The interest in, and challenge of, making consistent and profitable use of order book data in our project mirrors similar interest on Wall Street.

The prevalence of order book strategies in PPC 2003 in part derives from an early sample strategy we provided to participants, known as the *Static Order Book Imbalance* (SOBI) strategy. This strategy computes volume-weighted average prices (VWAP) of (quartiles of) the PXS buy and sell order books, and compares them to the PXS last price. The basic idea is that if, for example, the VWAP of the buy book is much further from the last price than the VWAP of the sell book, it is a sign that buyers are less supportive of this price than are sellers, as indicated by their limit orders (statistically) standing further off. In this case, SOBI will place an order to sell shares, on the theory that the weaker buy-side support will cause the price to fall. SOBI has a number of parameters, including what fraction (quartile) of the books to include in the VWAP computations, how strong the VWAP imbalance must be in order to trade, the volume of shares to trade, whether to place market or limit orders and at what price, and some simple hedging mechanisms. It is an example of a simple strategy that interprets limit order books as an expression of market sentiment, and would not have been possible to implement just a few years ago, before order book data became available to automated and human traders.

A couple of the strategies in PPC 2003, such as CBR-SOBI and CReaTiv, can be broadly viewed as significant modifications of the basic SOBI idea, in the former case by adding an interesting learning mechanism for parameter adjustment, and in the latter by taking real-time measures of volatility into account. In addition, there were several other strategies that chose to use order book data in different ways, including RaSTa and OBSigma. Machine learning techniques applied to feature vectors derived from order books was also a significant component of two strategies — case-based reasoning in the case of CBR-OBI, and Boosting in the case of client CIA. Overall, it appears that 9 of the 14 entrants made use of order book data in some way, though the centrality of this data to the strategy varied considerably.

The entrants also included a number of clients that implemented variations on more traditional technical trading strategies, such MoneyFlow, OBCrossover and OBBreakout. There were also several clients that implemented some form of market-making strategy (OBMM, DAMM-STAT and RapidMM).

Overall, the entrants permitted the creation of two diverse pools of interesting strategies, varying from the extremely simple (such as Contrarian) to the rather complex, and exhibiting a range of trading styles that includes both those commonly found on Wall Street today (such as market-making and certain technical trading methods), and rather new (and untested) methods, such as those relying heavily on order book data. We have deliberately encouraged this diversity throughout the project, believing it increases the interest and realism of the simulations. Of course, the results of PPC 2003, which we now summarize and analyze, are strongly dependent on the particular set of clients, and even on the specific division into pools, as we shall see shortly.

4.2.3 Summary of Results

Figure 5 contains a summary of the overall performance for each client over the 10 days of PPC 2003. Clients are sorted by their pool, and then listed in order of their final ranking according to the scoring criteria of Figure 3. In addition to the overall point totals that determined the final standings, subtotals for each of the 6 point categories are given as well.

The winners of PPC 2003 were the clients CBR-SOBI (Blue pool) and DAMM-STAT (Red pool). It is striking how differently these strategies managed to emerge as the victors in their respective pools. While CBR-SOBI was also among the top performers in terms of raw profitability (discussed shortly), and earned the majority of its points in categories directly related to positive earnings, client DAMM-STAT was barely profitable overall, but managed to succeed by consistent adherence to good trading practices, managing to pick up the maximum of 20 points for risk saturation, and earning 14 points for unwinding its position on 7 of the 10 days. The next best performance in unwinding in either pool earned only 6 points. Overall, we were quite pleased with the balance between profitability, consistency and good trading practices that the scoring criteria brought out among the better performers. While many of the lower-ranked clients had positive earnings, they all consistently failed in one or more of the basic practices or behaviors we encouraged.

In terms of profitability, 11 of the 14 clients ended with overall positive cumulative earnings for the 10-day competition. Of course, just as on Wall Street, we must consider the question of both the statistical significance of earnings, and the trade-off between risk and return. A common measure of the latter is known as the *Sharpe Ratio*, which is the empirical daily average of returns divided by the standard deviation. The ideal, of course, is to have a large Sharpe ratio — consistently high earnings with very small spread in the returns. Among the 14 clients, there were two (CBR-SOBI and OBVol) that achieved noteworthy Sharpe ratios, and one (SimpleTrend) that actually displayed a strongly *negative* Sharpe ratio. Despite the strong monetary performance of OBVol, it fared less well by the scoring criteria, primarily due to a consistent failure to saturate the allowed risk in comparison to the higher-ranked clients, which all received all 20 points in this category.

While accounting for the risk-return trade-off, the Sharpe Ratio is insensitive to the amount of data available, and thus is generally not reliable as an indicator of statistical significance. We thus also provide 95% confidence intervals around the average for each client. By this measure, there were two clients (again CBR-SOBI and OBVal) whose confidence intervals lie exclusively in the the region of positive earnings, and thus pass this standard test for statistical significance at the 0.05 level on just 10 days of data. By the same token, we can assert that client SimpleTrend is, with high confidence, a money-losing strategy.

Overall, the PPC 2003 scoring criteria seemed to effectively balance profitability considerations with our other interests. More precisely, the correlation coefficient between the client point totals and their profit and loss totals was 0.41, and thus the importance of profitability was considerable without being dominant. This is not surprising considering the motivation behind the design of the criteria.

In addition to the above overview of client performances, there were many more detailed aspects of the competition that are worthy of discussion, which we now provide.

4.2.4 Analysis

We begin the analysis with an examination of one of the most coarse measures of overall market activity, the total volume of shares matched by the simulator during each trading day. Figure 6 shows, for both the Blue and Red pools, the 10 plots of the total number of shares matched by PXS for each of the 10 competition days. The differences in behavior between the two pools are striking, with the end-of-day matched volume varying between 4 and 13 million shares traded in the Blue pool, and between 4 and 8 million shares in the Red pool. Thus, the Blue pool saw considerably greater total volume, as well as greater variability in volume.

The evolution of volume traded with time shown in the plots also reveals that Blue pool trading was prone to sudden rushes of trading activity, while Red pool volume grows relatively smoothly throughout the day. We shall examine the causes of this Blue pool volatility shortly when we discuss more client-specific behavior.

It is instructive to compare these volume plots with those of the Island exchange itself. Figure 6 also shows the total matched Island volume with time for the same 10 days of the competition. End-of-day volumes vary from roughly 7 to 9 million shares; thus, the total trading activity overseen by PXS among the 7 clients in each pool rivaled or exceeded the entire volume traded by the external Island market. In this sense, it is fair to say that the competition clients were indeed very high-volume traders. (We note that although

Strat	Pool	Rk	***Tot***	***P***	***S***	***U***	***R***	***C***	***V***	**Av P&L ($)**	**95% Int**	**Sharpe**
CBR-SOBI	B	1	74	9	18	2	10	15	20	4187	± 3733	0.70
MoneyFlow	B	2	69	15	20	0	4	15	15	2007	± 15692	0.08
OBMM	B	3	46	8	20	0	8	0	10	258	± 7909	0.02
CReaTiv	B	4	42	7	20	0	10	0	5	(2410)	± 6770	(0.22)
OBCrossover	B	5	33	6	6	0	6	0	15	3242	± 4220	0.45
OBBreakOut	B	5	33	10	18	0	0	0	5	3680	± 7963	0.29
RaSTa	B	7	21	5	2	0	4	0	10	1182	± 2441	0.30
DAMM-STAT	R	1	65	6	20	14	10	0	15	685	± 5195	0.08
Contrarian	R	2	55	6	20	2	12	0	15	2022	± 3658	0.34
OBSigma	R	3	54	8	20	6	10	0	10	1649	± 2382	0.43
OBVol	R	4	53	14	0	0	4	15	20	4037	± 1900	1.32
RapidMM	R	5	50	10	20	0	10	0	10	3649	± 9121	0.25
CIA	R	6	30	13	12	0	0	0	5	(1451)	± 9822	(0.09)
SimpleTrend	R	7	27	3	20	2	2	0	0	(24467)	± 17974	(0.84)

Figure 5: Summary of Results for PPC 2003. Column ***Tot*** shows total points for each client over the 10-day competition. Column ***P*** shows points earned in the Daily Profit and Loss category. Column ***S*** shows points earned in the Daily Risk Saturation category. Column ***U*** shows points earned in the Daily Position Unwinding category. Column ***R*** shows points earned in the Daily Intraday Position Reversal category. Column ***C*** shows points earned in the Overall Consistency of Profit and Loss category. Column ***V*** shows points earned in the Robustness to Market Variation category. Column **Avg P&L** shows average daily Profits and Losses, while column **95% Int** gives the 95% confidence interval around this mean. Column **Sharpe** gives the 10-day Sharpe Ratio for the client. See text for more details.

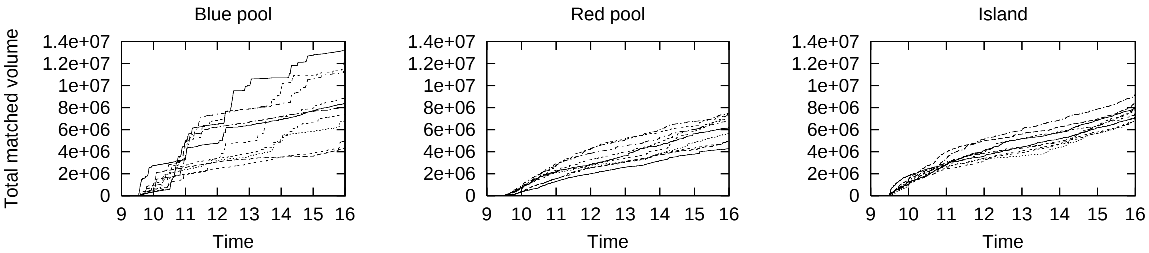

Figure 6: Total matched PXS volume as a function of time for the Red and Blue pools, and total volume on Island, over the 10 competition days.

all of the shares exchanged by Island are *available* for trading in PXS via Step 4 of the execution engine description in Figure 2, this does not necessarily mean that such shares are consumed in the PXS market.) The shape of the Island volume plots is more similar to the smooth Red pool behavior, but also demonstrates the well-known fact that real-world trading activity tends to be greater shortly after the open and shortly before the close.

Recall that one of the main motivations behind the design of PXS and the competition is the creation of a hybrid market that allows a diverse market of virtual clients to also be influenced or "corrected" by the incoming stream of real market data. We are thus naturally interested in examining the extent to which the internal PXS market tracked or deviated from the Island market. Overall, the competition seems to have been quite successful in balancing the influence of the virtual and external market. A typical plot of the PXS and Island last prices generally shows that over the entire trading day, the two prices are extremely close, with occasional short-term deviations of small magnitude (some notable exceptions to this are discussed shortly).

More quantitatively, in Figures 7 and 8, we show histograms of the differences between Island last price and PXS last price, aggregated over all 10 competition days. The greatest mass in these histograms lies very close to 0 cents, and virtually all of the mass is contained in a margin of ± 3 cents. It thus seems that the external influence of the Island data did indeed cause a close, but not perfect, correlation between the two markets. Note the symmetry of the histograms around 0, indicating that PXS deviations from Island price are unbiased, as one might expect.

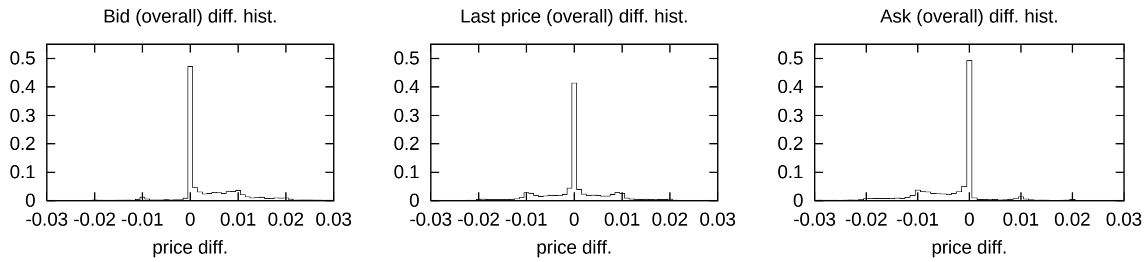

Figure 7: Histograms of differences between Island and Blue pool PXS markets, aggregated over the 10 competition days. The middle figure shows the histogram of values for *(PXS last price - Island last price)*, the left figure for *(PXS bid - Island bid)*, and the right figure for *(PXS ask - Island ask)*.

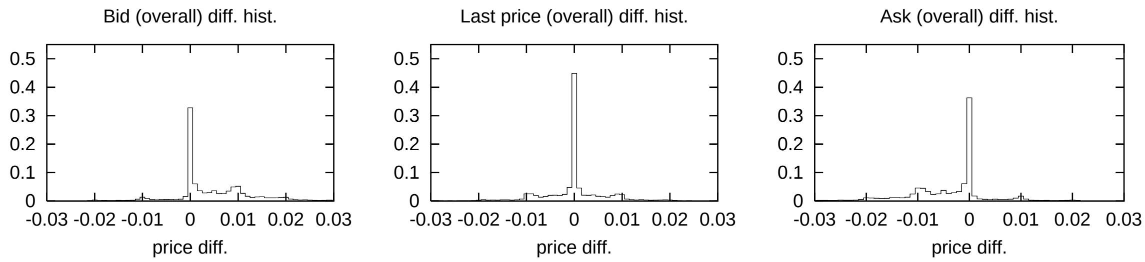

Figure 8: Histograms of differences between Island and Red pool PXS markets, aggregated over the 10 competition days. The middle figure shows the histogram of values for *(PXS last price - Island last price)*, the left figure for *(PXS bid - Island bid)*, and the right figure for *(PXS ask - Island ask)*.

The same figure also gives histograms of the differences between Island and PXS bid and ask prices (the inside market). Here we again see a close correspondence, with the mass entirely contained in a margin of ± 3 cents. Now, however, the histograms are asymmetric: the PXS bid price was much more frequently above the Island bid, and the PXS ask much more frequently below the Island ask. In other words, the PXS inside market was generally tighter than Island's, a sign of greater liquidity and competition in the internal market.

It is also interesting to examine coarse measures of the differences in the overall books between PXS and Island. In Figure 9 we plot, for each of the 10 days of the competition, the total number of *orders* in the Island buy and sell books as a function of time. Interestingly, the Island buy book seems to regularly exceed the Island sell book by about 100 orders; the reason for this is not clear. The Island books, while fluctuating somewhat during a typical day, seem to maintain roughly constant size during normal trading hours.

In contrast, as can be seen in Figures 10 and 11, the number of orders on the PXS books showed considerably more volatility, and there were again significant differences between the pools. In the Blue pool, there were frequent crossing between the number of buy versus sell orders, while in the Red pool, buy orders usually outnumber sell orders, and there are frequent sudden rushes of buy orders. This latter behavior may well be the result of the actions of a single client.

We now turn to an analysis of more client-specific behaviors during the competition. Perhaps the easiest way to visualize the overall trading behavior of individual clients is to examine their share positions throughout the trading day. Figures 12 and 13 contain one set of axes for each of the 7 clients in both the Blue and Red pools, with each client axes containing 10 plots showing the number of shares (long or short) held by the client as a function of time over the 10 competition days. Although the crowded nature of the plots makes it difficult to track the precise position of a client on any single day, we can infer from these plots a great deal of macroscopic information about the client's frequency and volume of trading, the bias towards long or short positions, and many other properties.

For instance, within the Blue pool, we see that the victorious client CBR-SOBI tended to execute relatively large (but still moderate) transactions at relatively long intervals, in comparison to a client such as OBMM, whose position plots increase or decrease in much smaller increments, but much more frequently. We can also see CBR-SOBI's frequent reversal of position within a trading day, and an overall balance between long and short selling. OBBreakout, on the other hand, shows an incapability of anything but overall short positions over the 10 days. The dark mass of heavy trading activity by client CReaTiv between approximately 10 AM and 11 AM each day is probably at least partially, if not primarily, responsible for the heavy increases in overall PXS volume that occur at about that time in the Blue pool on several trading days, as seen in Figure 5.

Notable client behaviors among the Red pool include the overall balance in volume, frequency and position demonstrated by the winning client DAMM-STAT; the heavy transaction rate of Contrarian, who also seems to halt trading by 3 PM each day; and the bias towards short and long positions, respectively, of clients OBVol and CIA (who also failed to trade significantly on several days).

Client SimpleTrend of the Red pool is worthy of special mention, as its behavior nicely demonstrates the fact that an internal pool of diverse and aggressive virtual clients can act as a proxy for the market impact that excessively large orders typically have in the real world. Unlike all other clients in either pool, SimpleTrend often engages in sudden transactions (both buying and selling) for close to a hundred thousand shares, as can be seen from its position plots. These large deals proved disastrous for SimpleTrend, and profitable for its trading partners, as the huge orders simply ate deep into the opposing book, and left SimpleTrend with progressively worse prices. (Recall that SimpleTrend had the worst P&L performance of the competition, and actually passes a statistical significance test for unprofitability.) Despite the previously discussed statistical closeness of the Island and PXS markets, the behavior of SimpleTrend caused the Red pool last price to instantaneously

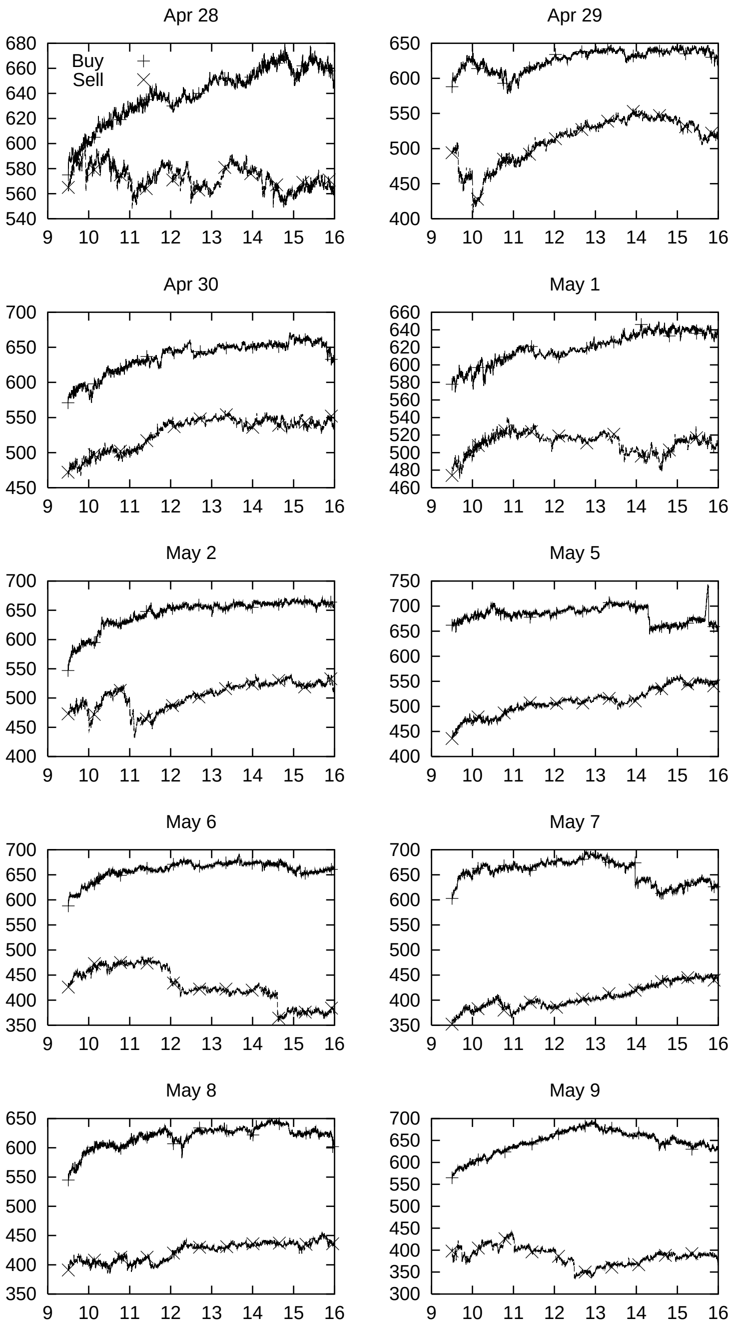

Figure 9: Island buy and sell books sizes, in number of orders, as a function of time. Each subplot shows the buy and sell book sizes for a different day of the competition.

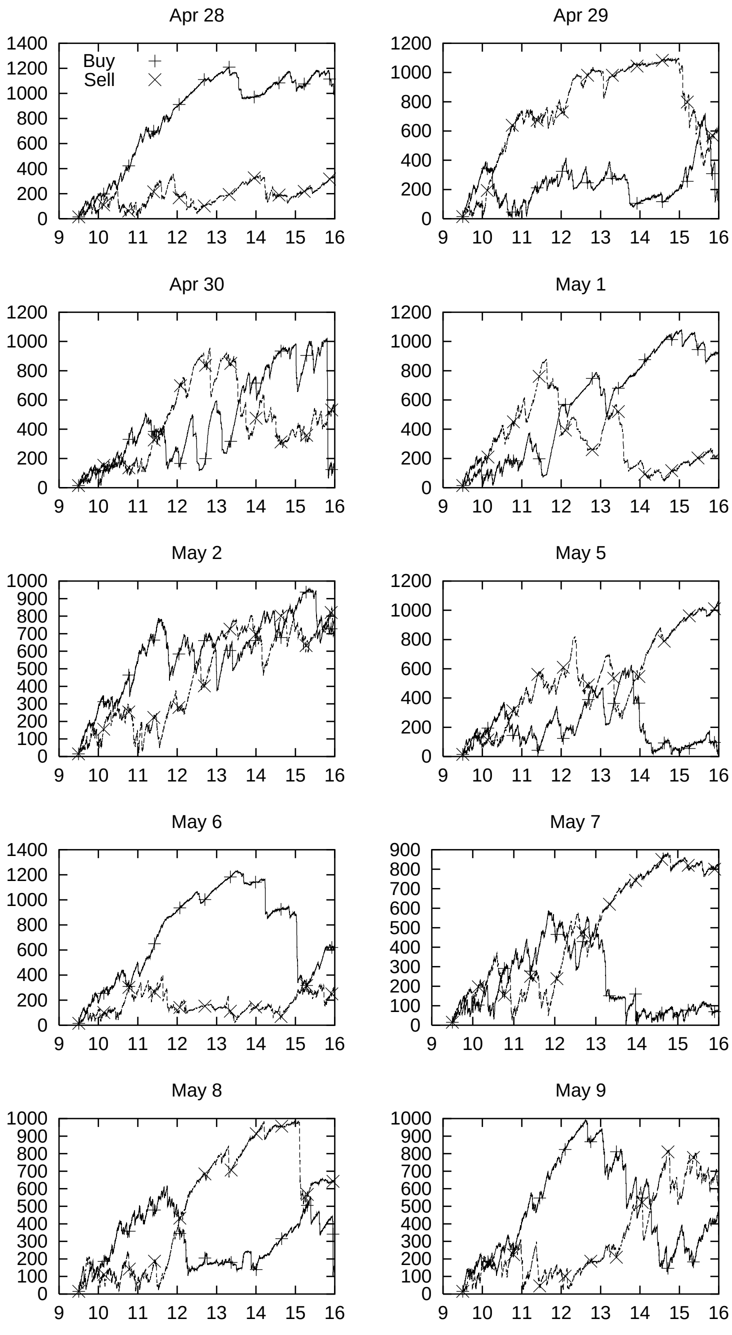

Figure 10: PXS Blue pool buy and sell books sizes, in number of orders, as a function of time. Each subplot shows the buy and sell book sizes for a different day of the competition.

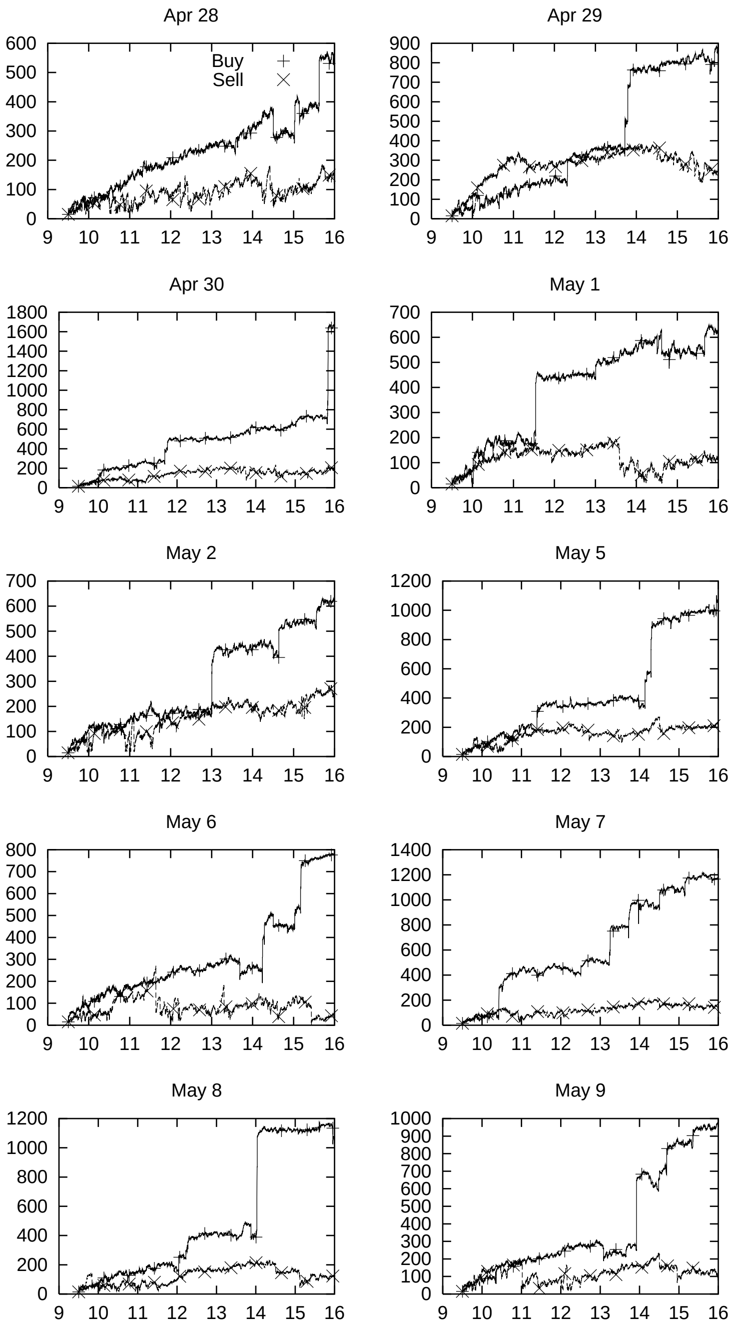

Figure 11: PXS Red pool buy and sell books sizes, in number of orders, as a function of time. Each subplot shows the buy and sell book sizes for a different day of the competition.

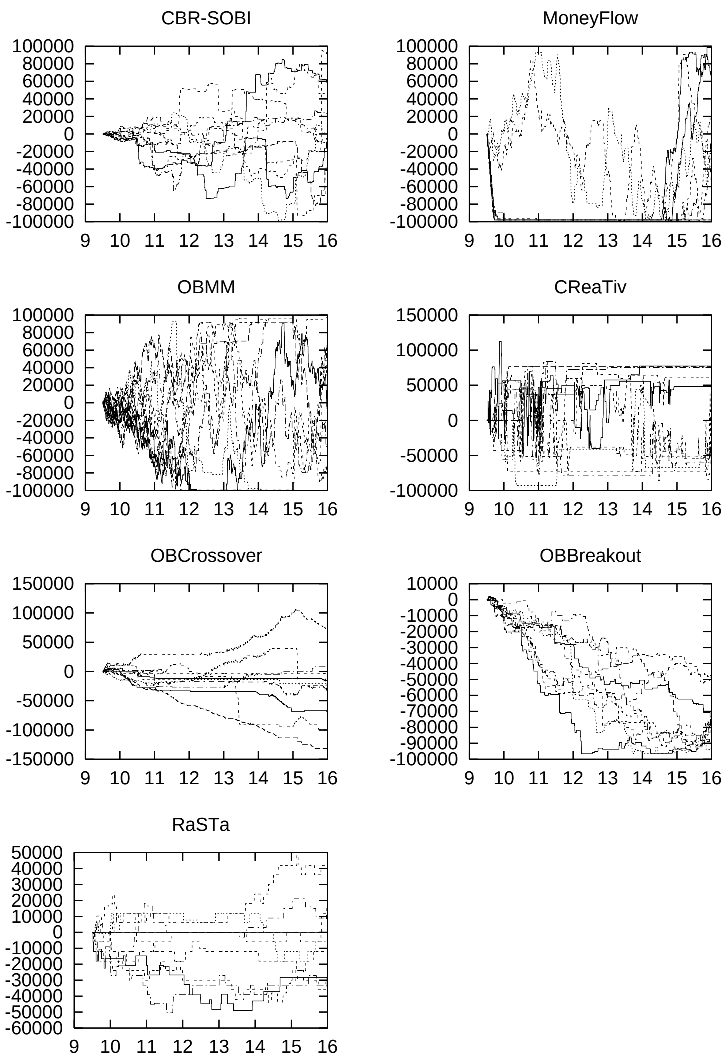

Figure 12: Blue pool client share positions with time. Each subplot shows, for the indicated Blue pool client, the number of shares owned or owed by the client as a function of time, with each subplot containing 10 graphs corresponding to the 10 days of the competition.

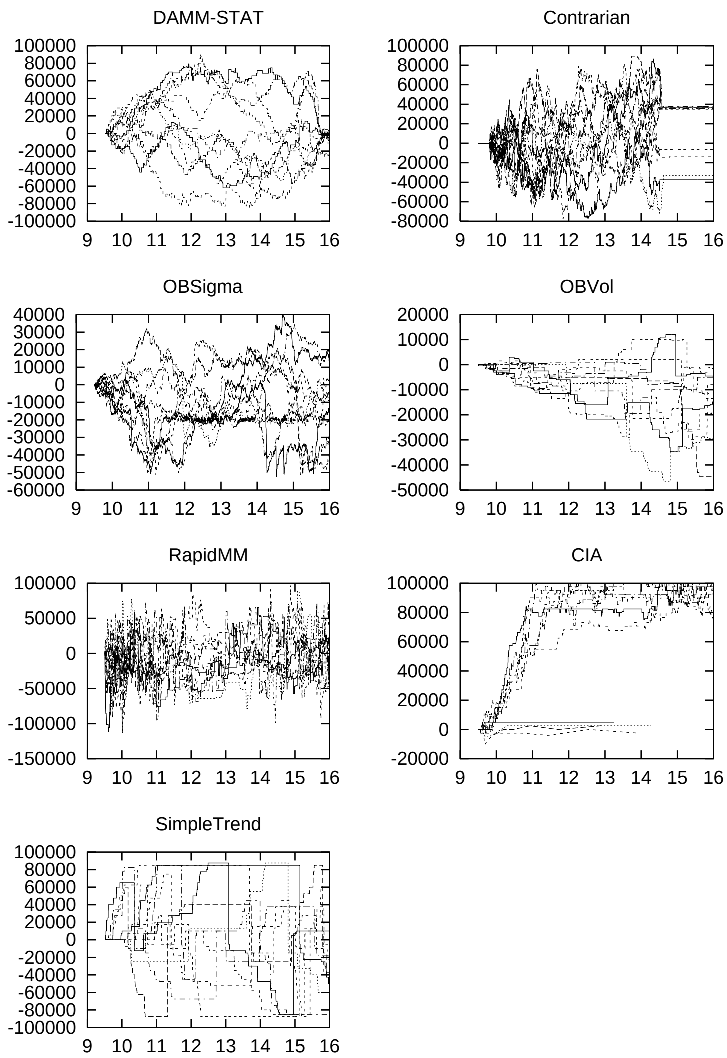

Figure 13: Red pool client share positions with time. Each subplot shows, for the indicated Red pool client, the number of shares owned or owed by the client as a function of time, with each subplot containing 10 graphs corresponding to the 10 days of the competition.

deviate dramatically from Island's several times during the competition; an example is shown in Figure 14, where the sudden changes in PXS price at three distinct moments during the day are directly caused by large SimpleTrend orders. Figure 14 shows a typical day for SimpleTrend, where several precipitous tumbles deeper into the red are directly aligned in time with the large position changes.

A complementary view of client behaviors is given in Figures 15 and 16, which are client-specific analogues of Figures 7 and 8, and show the overall matched volume for each client versus time over the 10 days. Overall differences in typical end-of-day volumes are apparent, with client such as OBMM and CReaTiv (Blue pool) and Contrarian and RapidMM (Red pool) being among the heaviest traders, often exchanging several million shares in a day, while many other clients traded only in the low hundred thousands. We also see notable differences in trading schedules and variation, with clients such as the Red pool's DAMM-STAT and OBSigma showing great regularity and roughly constant transaction rate across days, while RapidMM and CIA show increased activity early and late each day. It is interesting to note that client SimpleTrend is in fact *not* among the heaviest providers of liquidity in the red pool, but actually has relatively modest overall volume. It was the extremely bursty manner in which SimpleTrend chose to deploy its limited volume that led to its poor performance, not the amount of volume itself.

Overall, while our small pool of virtual clients probably deviates from typical Wall Street traders (even automated ones) in the large volume of trading they engage in, we were pleased with the realism of the PXS market, the healthy influence of the external Island data, the corrective effects of the other virtual clients in the case of SimpleTrend, the liquidity and tightness of the PXS market, and the diversity of the client population.

5 Ongoing Work and Conclusions

The Penn-Lehman Automated Trading Project is a work in progress, and we are actively planning on extensions to all aspects of the project. On the systems and platform side, we are currently enhancing the client API, improving the speed and robustness of the internal algorithms in the PXS execution engine, and designing a web-based GUI that will permit remote participants to use PXS and join our competitions. On the strategy side, we are using PXS to investigate a range of order book-based trading algorithms.

We actively solicit external participation in the project from researchers in both academic and industrial settings. Interested parties should contact M. Kearns at mkearns@cis.upenn.edu.

Acknowledgements

Special thanks and acknowledgement to Berk Kapicioglu and Byoungjoon Kim, who did the initial development and testing of PXS. Thanks also to Narayanan Mahesh, Nick Montfort and Rashid Tuweiq, who made numerous contributions to the system.

We are extremely grateful to the Proprietary Trading Group at Lehman Brothers in New York City for their generous scientific, professional and financial support of the project. Author M. Kearns would like to give special thanks to his Lehman colleagues Michael Bleich, Michael Bos, Andy Ellner, Amy Papandreou, Colin Rust, and Mark Sanborn for their time and knowledge.

We also thank all of the participants in the PLAT project who developed PXS clients for their hard work and patience. We are particularly grateful for the suggestions and comments of Yuriy Nevmyvaka of Carnegie Mellon University and Peter Stone of the University of Texas at Austin.

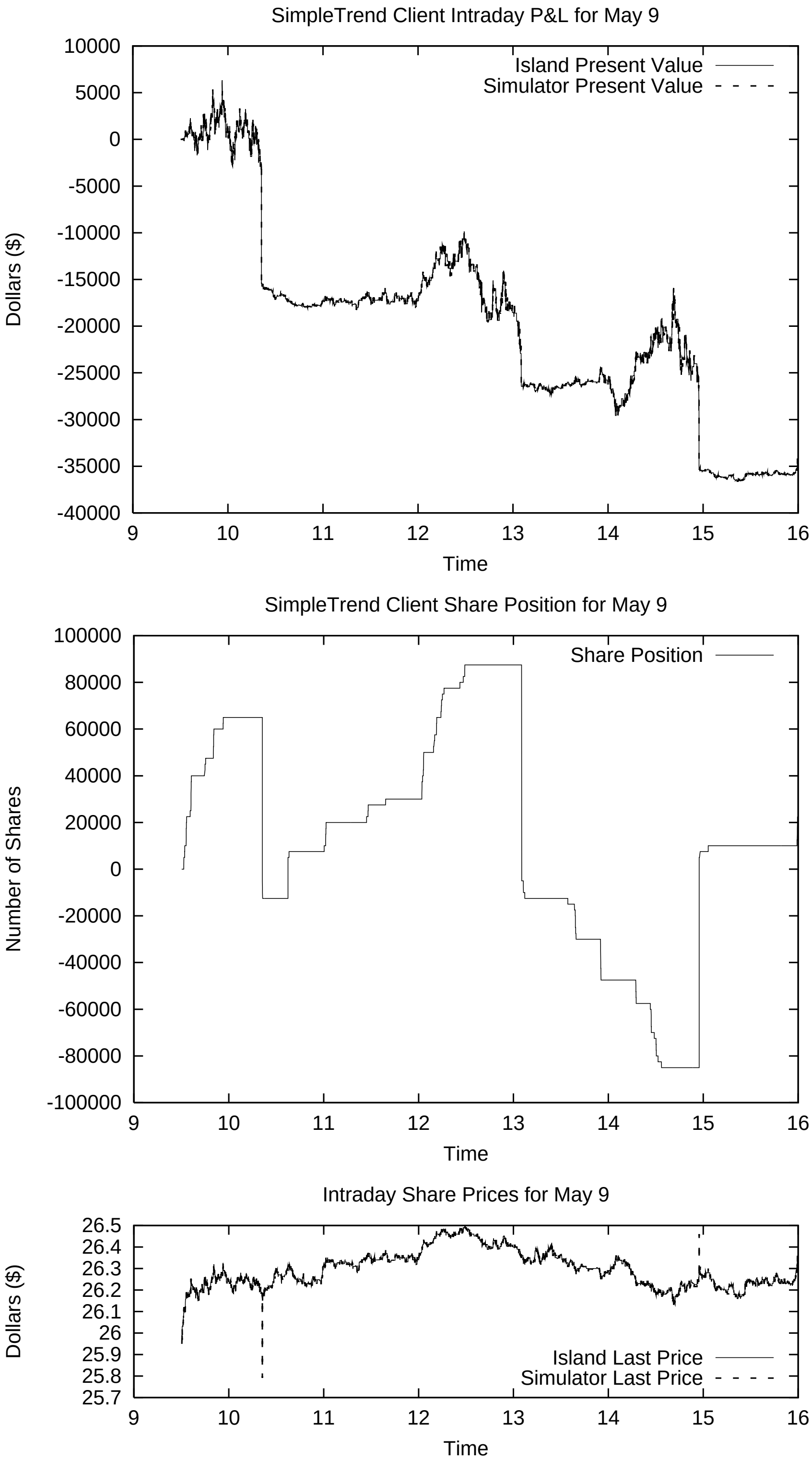

Figure 14: Plots of May 9 P&L (top) and share position (middle) for Red pool client SimpleTrend. Note that the execution of three trades in excess of 60,000 shares each (a sell shortly after 10 AM, a sell around 1 PM, and a buy around 3 PM) are each accompanied by sharp losses, as the large orders eat deep into the PXS order books. The bottom plot shows a comparison of the PXS and Island last prices over the same trading day; while indistinguishably close for almost the entire day, the first and last of SimpleTrend's large trades cause instantaneous and large deviations of the PXS price in the corresponding direction. At the large trade around 1 PM, there was apparently enough depth in the PXS books near the inside market to absorb the trade, and prevent a deviation between Island and PXS prices.

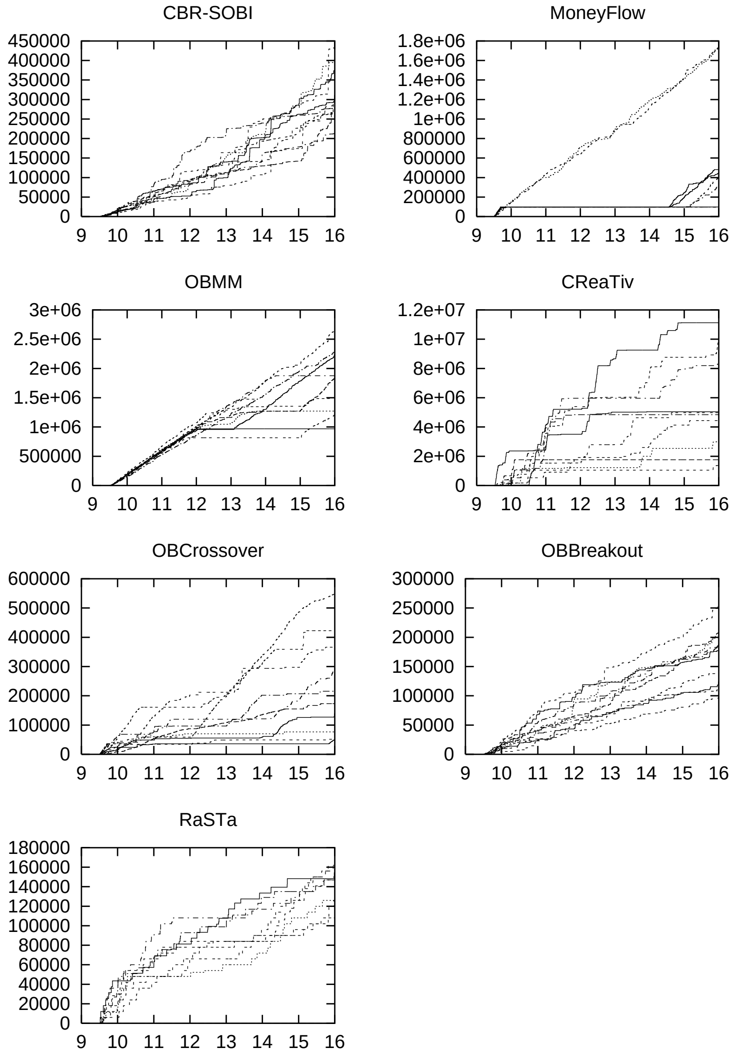

Figure 15: Blue pool client matched volume with time. Each subplot shows, for the indicated Blue pool client, the total number of shares matched (bought or sold) as a function of time on the 10 days of the competition.

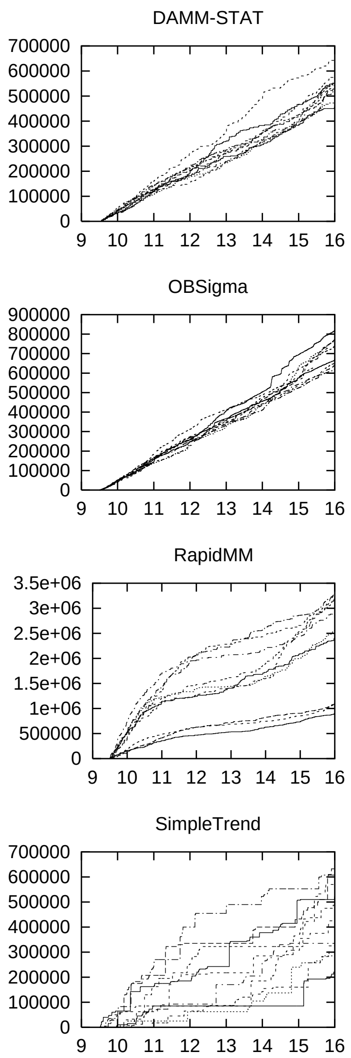

Figure 16: Red pool client matched volume with time. Each subplot shows, for the indicated Red pool client, the total number of shares matched (bought or sold) as a function of time on the 10 days of the competition.

References

[1] *Proceedings of the Fourth Conference on Electronic Commerce*. Association for Computing Machinery, 2003. Forthcoming. Conference web site: http://cs.gmu.edu/~menasce/ec03/s/ec03.htm.

[2] J. Bouchaud, M. Mezard, and M. Potters. Statistical properties of stock order books: Empirical results and models. *Quantitative Finance*, 2, 2002.

[3] J. Hasbrouck and G. Saar. Limit orders and volatility in a hybrid market: the island ECN. 2002. NYU Finance Department Working Paper.

[4] P. Stone, M. Littman, S. Singh, and M. Kearns. ATTac-2000: An adaptive autonomous bidding agent. *Journal of Artificial Intelligence Research*, 15, 2001.

[5] M. Wellman, A. Greenwald, P. Stone, and P. Wurman. The 2001 trading agent competition. *Electronic Markets*, 13(1):4–12, 2003.

[6] M. Wellman, P. Wurman, K. O'Malley, R. Bangera, S. Lin, D. Reeves, and W. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, 2001.