The Penn-Lehman Automated Trading Project

Michael Kearns and Luis Ortiz, *University of Pennsylvania*

The PLAT Project has developed a trading simulation that merges automated clients with real-time, real-world stock market data. This simulation has been used for three competitions.

The Penn-Lehman Automated Trading Project is a broad investigation of algorithms and strategies for automated trading in financial markets. The PLAT Project's centerpiece is the Penn Exchange Simulator (PXS), a software simulator for automated stock trading that merges automated client orders for shares with real-world, real-time order data. PXS automatically computes client profits and losses, volumes traded, simulator and external prices, and other quantities of interest. To test the effectiveness of PXS and of various trading strategies, we've held three formal competitions between automated clients.

PLAT background

The PLAT Project (www.cis.upenn.edu/~mkearns/projects/plat.html) has several underlying motivations. From a research perspective, we're among the growing number of AI and computer science researchers with an interest in all forms of e-commerce, computational markets, algorithmic mechanism design and electronic auctions, and related topics. In addition to a burgeoning theoretical literature,[1] this line of research has a growing platform and systems component. The best example of such a system is perhaps the popular, successful Trading Agent Competition (TAC)[2–4] (see also http://auction2.eecs.umich.edu/researchreport.html), which has focused primarily on multicommodity auction simulations. So, one primary motivation for the PLAT Project is to contribute 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're also interested in designing challenging, realistic competitions in automated trading in financial markets, using PXS as the testbed.

We also actively use PXS as a platform for developing novel, principled automated trading strategies (clients). The real-data, real-time nature of PXS lets us examine computationally intensive, high-frequency, high-volume trading strategies (although this last property always presents the challenges of estimating the *market impact*—the effect on prices). We're particularly interested in developing clients that make predictive use of limit order book data, including those using statistical modeling and machine learning. We hope that, over time, the project will generate a library of clients with varying features (trading strategy, volume, frequency, and so on) that can serve to create realistic simulations with known properties.

In addition, the project has a major educational component. Aside from the project staff team of five (who oversee PXS maintenance and development, manage the Island data, and run the competitions), over 30 students are developing automated trading strategies for PXS, and they regularly participate in the competitions. Many of the students are in joint programs with the University of Pennsylvania's Computer and Information Science Department and the Wharton School (of business). These students must undertake a year-long senior research project; many have chosen to do so as participants on the PLAT Project developing novel automated-trading strategies. Several students are from other universities.

Finally, the PLAT Project is an educational and institutional partnership between the University of Pennsylvania and Lehman Brothers' Proprietary

1094-7167/03/$17.00 © 2003 IEEE
Published by the IEEE Computer Society

Trading Group in New York City, a group of Wall Street professionals who actively design and implement a rich, 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 acted as mentors to the participating students.

Market microstructure

To understand the PLAT Project, as well as PXS, you need to understand some details of the underlying mechanics of financial markets and exchanges. These computational and transactional details are sometimes broadly called *market microstructure*. We focus on the market microstructure of NASDAQ exchanges and stocks, both for specificity's sake and because it's most directly related to PXS. However, many of the same elements appear in other exchanges, including the New York Stock Exchange.

Unlike the NYSE, NASDAQ is entirely electronic. All orders, whether an algorithm or a person generates them, are sent to NASDAQ via an electronic interface and order-routing system. Also, a computer executes all matches between buyers and sellers. 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 stock's *specialist*), NASDAQ lets many firms and individuals provide markets in NASDAQ stocks. This environment led to the advent of *Electronic Crossing Networks* (also called Electronic Communication Networks), which are firms providing essentially independent, competing markets for NASDAQ stocks. A typical NASDAQ stock trader will track prices and activity simultaneously in multiple ECNs. He or she might either break large orders up over several of them or prefer certain ECNs for certain types of transactions, depending on differing fee structures. Recently, some mergers and institutional efforts toward consolidation have occurred, but several large, independent ECNs still exist, including Island (www.island.com), Instinet (www.instinet.com), and Archipelago (www.archipelago.com).

A fundamental distinction in stock trading is that between a *limit order* and a *market order*. Suppose we wish to purchase 1,000 shares of Microsoft (whose NASDAQ ticker symbol is MSFT) stock. In a limit order, we specify not only the desired volume but also the desired price. Suppose that MSFT is trading at roughly $24.07 a share (see Figure 1), but we only want the 1,000 shares at $24.04 a share or lower. We can submit a limit order with this specification, and our order will be automatically 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 called the *bid*). If multiple limit orders have the same price, they are ordered by arrival time (with older orders higher in the book). In the example in Figure 1, our order would be placed immediately after the extant order for 5,503 shares at $24.04. Although we offer the same price, this order 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 called the *ask*).

So, the order books are automatically 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 together are sometimes called the *inside market*, and the difference between them is the *spread*. By definition, the order books always consist exclusively of unexecuted orders—they're queues of orders waiting for the price to move in their direction.

How then do orders get executed? By definition, when a market order arrives, it's immediately matched with the most competitive limit orders on the opposing book. So, a market order to buy 2,000 shares will be matched with enough volume on the sell order book to fill the 2,000 shares. For instance, for the example in 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 1,981 shares at $24.09. The remaining 1,681 shares of this last limit order would remain as the new top of the sell limit order book. A limit buy order with a price much higher than the current ask is effectively a market order. Likewise, a limit sell order with a price much lower than the current bid is effectively a market order. For this reason, some ECNs (including Island) don't offer a separate market order mechanism. But conceptually, the two types have an important difference: a limit order is guaranteed price (if executed) but not execution, whereas a market order is guaranteed execution but not price (because the books might change before the order arrives at the exchange and executes).

MSFT

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

Figure 1. A typical order book for an electronic crossing network.

In this setting, every market or limit order arrives atomically and instantaneously—orders arrive in a strict temporal sequence, and two orders can never arrive simultaneously. This gives rise to the definition of the exchange's *last price*, which is simply the last price at which the exchange executed an order. This quantity is what people usually mean when they casually refer to a stock's (ticker) price. The last price might change more slowly than the order books, especially in less liquid stocks.

The market microstructure we've described has existed since the dawn of financial markets, with some variations in the details (for instance, NYSE specialists have some flexibility in how and when they execute matches between buyers and sellers). What's more recent is the automation of this process in markets such as NASDAQ. The fundamental role of ECNs (and PXS) is the computerized maintenance of buy and sell order books in the offered stocks, automated order execution, and various other related functionalities (such as withdrawing or changing unexecuted orders or checking a previously placed order's status).

Even more recent than the automation of market microstructure is the publication of real-time order book data. The publication of such data presents intriguing opportunities (it

has also spawned a relatively new literature attempting to come to both a theoretical and empirical understanding of order book behavior[5,6]). This is because the limit order books can be viewed as an expression of market sentiment and, more prosaically, might provide strategic guidance for order placement. Indeed, Wall Street traders commonly 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).

The availability of order book data makes PXS possible. More specifically, Island is a major ECN for NASDAQ stocks, accounting for approximately one of every seven NASDAQ trades. It's thus among the most important providers of liquidity in NASDAQ, and like many ECNs, it's a technologically sophisticated company. Besides accepting orders arriving through a variety of standard Wall Street brokerage order-routing systems, it provides an API for automated order placement and management.

The Penn Exchange Simulator

PXS is basically an experimental ECN that merges limit order data from two sources: actively connected trading clients and limit orders from Island. We obtain the Island data in real time from Island's live Web-based BookViewer, which shows the top 15 limit orders (price and volume) in the buy and sell order books. (Because BookViewer shows only the top 15 orders rather than the entire books, accurate maintenance of the PXS books involves some technical challenges. We discuss these challenges in an expanded version of this article, available at www.cis.upenn.edu/~mkearns/papers/plat.pdf.)

PXS simulations run in one of two modes. Live mode updates the Island data in real time from BookViewer approximately every three seconds. Historical mode requires that the requested day of Island data has already been archived. Live simulations provide the most realistic merging of the Island and PXS markets because they're faithful to the timing details of activity on Island. Historical simulations are considerably faster because PXS can process each update of Island data at its own internal processing rate.

Every major Wall Street brokerage has multiple platforms and data sources on which to test new trading strategies; these are often called *backtesting* environments. To date, most or possibly all backtesting environments employ either price information alone or, in some cases, price and inside-market data. This is almost certainly because order book data has been widely available for NASDAQ only very recently (and is still not yet available in many major markets).

One major advantage of an order-based simulation platform such as 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 will execute at a future time based on only that information. Obviously, if the price never reaches the limit order price, the order will never get filled. But, if the price does cross the limit order price, it might or might not have 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 on the basis of the historical data and the order's volume.

Just as in real exchanges, PXS fills a limit order only if an opposing order matches it 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.

We implemented PXS in C on Unix and Linux platforms. When it's invoked, it takes these arguments:

- The four-letter ticker symbol of the Island-traded NASDAQ stock for which to run a simulation (such as MSFT).
- The port number over which PXS will communicate with trading clients.
- For historical simulations, the date for which the simulation should run. If this argument is omitted, PXS executes a live simulation using the current day's Island data.
- For the trading day being simulated (whether live or historical), the simulation's start and stop times.

For example, the command

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

starts an execution of PXS on Island MSFT order book data from 29 April 2003 at 9:30 am and terminates 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 without the -h argument immediately starts the simulation, using the current Island MSFT data.

The 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. To each connecting client, PXS assigns a client identifier.

The PXS client API 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 we discuss here is split between PXS and a client shell process inside which participants implement their trading strategies. Generally, PXS computes quantities of global interest to the simulation, while the client shell computes more client-specific quantities. For simplicity, we'll blur this distinction and simply refer to PXS.

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

- buyOrder(p, v) places a limit buy order at price p for v shares. It returns an identifying number for the order.
- sellOrder(p, v) places a limit sell order at price p for v shares. It returns an identifying number for the order.
- withdraw(o) withdraws the order with the identifier o from the PXS order books. This will fail if the order already has executed.

Some functions compute these agent-specific quantities of interest:

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

Some functions provide information about the overall PXS market state:

- The last price of the PXS or Island markets.
- The current time according to PXS. For live simulations, this will always be close to the actual (wall clock) time. For historical simulations, it will be the time stamp of the most recently processed Island update.
- The total volume of shares and number of orders in the PXS buy and sell order books.
- The total volume and number of orders that PXS has already matched in the simulation.

- Various statistics of the PXS order books, including volume-weighted average prices.

More generally, clients can receive a copy of the entire PXS order book data structure. This copy shows the price and volume of every limit order in the book, with indications of which orders came from Island and which from PXS clients. So, PXS provides clients with a level of internal market visibility that matches that offered on real ECNs such as Island. This lets PXS clients exploit market microstructure data at different levels of sophistication.

The execution engine

At the heart of PXS is the execution engine, which

- Maintains the PXS order books
- Integrates Island client buy and sell limit orders into these books
- Executes matching orders of PXS clients
- Computes PXS clients' share position, cash holdings, and profit and loss

A detailed description of the PXS execution engine is beyond this article's scope; for more information, see the expanded article at www.cis.upenn.edu/~mkearns/papers/plat.pdf. Here we simply give an overview and explain some of the engine's subtle aspects.

At a high level, the execution engine consists of five steps that execute repeatedly throughout a simulation:

1. Update the PXS buy and sell order books with any new orders detected on Island.
2. "Clean" the PXS books after the updates, to address our partial observability of the Island books.
3. Execute the PXS orders that are matched by executions on Island between updates.
4. Execute matches between buy and sell orders in the PXS books; update the PXS last price; and update client share positions, cash holdings, and profit and losses.
5. Insert the newly arrived client orders into the PXS books.

The valuation of holdings

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 can buy or sell shares, regardless of its current share and cash holdings. So, clients can sell more shares than they hold (selling short) or buy shares without cash. Share and cash holdings might 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 the sum of its cash balance (positive or negative) and its share position (positive or negative) times the stock's current (last) price. Thus, PSX makes an important "infinite liquidity" assumption: at any moment, any client could return its share position to zero instantaneously by placing all held shares on the market and receive the current price for all of them. For large share positions, this assumption is clearly unrealistic, because the immediate attempt to buy or sell large numbers of shares will move the price unfavorably. Further-

more, there's the choice of whether to use the Island or PXS last price to compute the valuation of share positions. PXS computes the valuations both ways. We further discuss valuation in the next section.

The competitions

Although PXS can perform simulations for any stock traded on Island (which includes all of NASDAQ), all three competitions involved Microsoft stock. Besides the students from the University of Pennsylvania, two teams from the University of Texas at Austin and one from Carnegie Mellon University have participated.

An issue common to all the competitions was risk management. Just as on Wall Street, it's important to encourage teams to develop strategies that intelligently balance their risk and return. In the project's setting, risk generally comes in the form of large share positions (long or short), which, as we mentioned before, are extremely vulnerable to unfavorable changes in the share price. So far, the project has focused exclusively on intraday trading; the competitions "cash out" each client at each day's end, with no positions held overnight. From this perspective, an ideal strategy would end each day with large cash holdings and a zero or small share position.

We wanted to prevent the results from being dominated by strategies that simply place large bets in the form of excessive share positions. So, all the competitions had the firm rule that, during any trading day, a client's share position must always remain within a window of ±100,000 shares. Violation of this was grounds for disqualification. We forgave minor infractions, but such a dismissal occurred in one of the early competitions. This limit is a crude and easily verifiable way of ensuring that no client succeeds simply by taking much larger positions than all the others. Because (at the time of the competitions) MSFT traded at roughly $25 a share, the share position limit effectively meant that clients would never have more than approximately $2.5 million of virtual capital at risk.

Another issue common to all 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 spent on purchasing shares throughout the day and the cash it received on sales). It will also have a long or short balance of MSFT shares. To convert this portfolio to a cash valuation, we must assign a value to the share position.

For large share positions, simply taking the share position and multiplying it by the stock's last price is unrealistically optimistic. This is because the placement of a market order for a large number of shares might eat deep into the limit order books, resulting in progressively less favorable prices. This is the well-known, difficult problem of assessing the market impact of large orders. It's also why accurate measurement of strategy performance from historical data is difficult and why brokerages almost always break large orders into small increments over time.

The presence of other trading clients in the PXS market can act as an important simulator of market impact (a topic we'll return to). However, 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 because we're trading a stock (MSFT) with high Island liquidity. Other valuation methods are possible, however, and we might explore them in future competitions.

The three competitions took place during November to December 2002, February to March 2003, and April to May 2003. In each competition, we divided the client strategies into pools, both for the sake of population diversity and to reduce each simulation's computational load. The first two competitions placed no restrictions on clients other than the share position limit, and we determined the winners strictly by their cumulative profitability. We won't discuss these early competitions' results in detail here, but one chief lesson we learned was that additional rules or scoring criteria were desirable to encourage increased realism among clients. This led to the third competition, dubbed the *Platinum Platter Competition*.

PPC 2003 divided the 14 entrant strategies into a Blue and a Red pool, each with seven clients. The division was somewhat arbitrary but took into account coarse preliminary experiments to ensure that each pool had a reasonable amount of client liquidity.

The competition occurred on each of the 10 trading days of the weeks of 28 April and 5 May. Each day's PXS simulation ran from 9:30 am to 4 pm, the normal NASDAQ trading hours. (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 of client evaluation, we used an interesting and considerably richer set of scoring criteria (see Figure 2) to encourage client realism and good trading practices.

The strategies

Table 1 briefly describes each PPC 2003 client and indicates its pool and its overall final pool ranking according to the scoring criteria in Figure 2. Many participants investigated variants of the Static Order Book Imbalance (SOBI) client that we provided for them. This client attempts to predict price movement on the basis of volume and price imbalances between the buy and sell order books.

One theme that emerged was how trading clients exploited order book information. Not only does PXS itself use 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, consistently and profitably exploiting order book data in our project mirrors similar interest on Wall Street. The clients CBR-SOBI and CReaTiv heavily used order book behavior and statistics. Two clients applied machine learning techniques to feature vectors derived from order books; CBR-SOBI used case-based reasoning, and CIA used the classification learning algorithm called *boosting*. Overall, nine of the 14 entrants exploited order book data; however, the centrality of this data to the strategy varied considerably.

Several clients, such as MoneyFlow, OBCrossover, and OBBreakout, implemented variations on more traditional technical trading strategies. OBMM, DAMMSTAT, and RapidMM implemented some form of market-making strategy, which seeks to profit from price volatility rather than overall movement.

Overall, the entrants formed two diverse pools of interesting strategies, varying from the extremely simple (such as Contrarian) to the rather complex. They exhibited a range of trading styles that includes both those commonly found on Wall Street, 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 proj-

ect, believing it increases the simulations' interest and realism. Of course, PPC 2003's results depend strongly on the particular set of clients and even on the specific division into pools, as you'll see.

The results

Table 2 summarizes each client's overall performance. The table sorts clients by their pool and their final ranking according to the scoring criteria of Figure 2. In addition to listing the overall point totals that determined the final standings, the table lists subtotals for each point category.

The PPC 2003 winners were CBR-SOBI (Blue pool) and DAMM-STAT (Red pool). It's striking how differently each strategy managed to emerge as the victor in its respective pool. CBR-SOBI was also among the top performers in terms of raw profitability (discussed shortly) and earned most of its points in categories directly related to positive earnings. DAMM-STAT was barely profitable overall but succeeded by consistent adherence to good trading practices. It managed to earn the maximum of 20 points for risk

Criteria emphasizing profitability

- *Daily profit and loss.* On a daily basis, award 3 points 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 (3×10) points.
- *Overall consistency of P&L.* A one-time award of 15 points goes to any client that has a positive cumulative P&L over the competition's 10 trading days and that ends with a negative daily P&L on three trading days at most. Maximum possible award: 15 points.

Criteria emphasizing robustness, with weak profitability prerequisites

- *Daily intraday position reversals.* On a daily basis, award 2 points to any client that finishes with a positive P&L for the trading day and that 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 (2×10) points.
- *Robustness to market variation.* Award 5 points to each client that has a 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, award an additional 5 points. Maximum possible award: 25 (5×5) points, if exactly 5 up days and 5 down days for the stock occur during the competition.

Criteria emphasizing good trading practices, with no profitability prerequisites

- *Daily risk saturation.* On a daily basis, award 2 points 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, award 2 points to clients whose total matched volume of shares for the trading day exceeds 1/14 (one-half of the per-client average of 1/7) of the total matched volume of all clients. Maximum possible award: 20 (2×10) points.
- *Daily position unwinding.* On a daily basis, any client that's awarded that day's risk saturation points can earn an additional 2 points by ending the trading day with a share position of fewer than 5,000 shares (long or short). Maximum possible award: 20 (2×10) points.

Figure 2. The client scoring criteria for the 2003 Platinum Platter Competition.

Table 1. 2003 Platinum Platter Competition clients.

Client	Description	Performance
CBR-SOBI	Case-based reasoning applied to the Static Order Book Imbalance strategy's parameters	First in the Blue pool; statistically significant profitability
MoneyFlow	A predictive strategy using money flow (price movement times volume traded) as a trend indicator	Second in the Blue pool
OBMM	A market-maker that positions orders in front of the *n*th 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 crossover strategy moderated by confirmation of order book quartile volume-weighted average prices	Tied for fifth in the Blue pool
OBBreakout	A breakout strategy applied to trend lines on the volume-weighted average prices of buy and sell books	Tied for fifth in the Blue pool
RaSTa	Resistance and Support Trading Agent—computes support and resistance levels on the basis of peaks in the order book volume	Seventh in the Blue pool
DAMM-STAT	A Mixture of a Dynamically Adjusted Market-Maker that 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 on the basis of relative spreads in the buy and sell books, interpreting a small standard deviation as a sign of confidence	Third in the Red pool
OBVol	A simple predictive strategy using total volumes in buy and sell books	Fourth in the Red pool; highest Sharpe ratio and statistically significant profitability
RapidMM	A market-maker with rapid revision of quotes based on the current inside market	Fifth in the Red pool
CIA	Central Intelligent Agent—a predictive strategy applying boosting to order book snapshots	Sixth in the Red pool
SimpleTrend	A simple trend prediction strategy	Seventh in the Red pool; statistically significant negative earnings

Table 2. PPC 2003 results.

Strategy	Pool	Rank	Total points	Points for: Daily profit & loss	Daily risk saturation	Daily position unwinding	Daily intraday position reversal	Consistency of P&L	Robustness to market variation	Avg. P&L ($)	95% confidence interval	10-day Shape ratio
CBR-SOBI	Blue	1	74	9	18	2	10	15	20	4,187	± 3,733	0.70
MoneyFlow	Blue	2	69	15	20	0	4	15	15	2,007	± 15,692	0.08
OBMM	Blue	3	46	8	20	0	8	0	10	258	± 7,909	0.02
CReaTiv	Blue	4	42	7	20	0	10	0	5	(2,410)	± 6,770	(0.22)
OBCrossover	Blue	5	33	6	6	0	6	0	15	3,242	± 4,220	0.45
OBBreakout	Blue	5	33	10	18	0	0	0	5	3,680	± 7,963	0.29
RaSTa	Blue	7	21	5	2	0	4	0	10	1,182	± 2,441	0.30
DAMM-STAT	Red	1	65	6	20	14	10	0	15	685	± 5,195	0.08
Contrarian	Red	2	55	6	20	2	12	0	15	2,022	± 3,658	0.34
OBSigma	Red	3	54	8	20	6	10	0	10	1,649	± 2,382	0.43
OBVol	Red	4	53	14	0	0	4	15	20	4,037	± 1,900	1.32
RapidMM	Red	5	50	10	20	0	10	0	10	3,649	± 9,121	0.25
CIA	Red	6	30	13	12	0	0	0	5	(1,451)	± 9,822	(0.09)
SimpleTrend	Red	7	27	3	20	2	2	0	0	(24,467)	± 17,974	(0.84)

saturation and earned 14 points for unwinding (that is, returning to zero) its position on seven of the 10 days. The next best performance in unwinding in either pool earned only six 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. Although 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 is to have a large Sharpe ratio—consistently high earnings with a very small spread in the returns. Among the 14 clients, two (CBR-SOBI and OBVol) achieved noteworthy Sharpe ratios, and one (SimpleTrend) actually displayed a strongly negative Sharpe ratio. Despite OBVol's strong monetary performance, it fared worse by the scoring criteria. Its poor score was due primarily to a consistent failure to saturate the allowed risk compared to the higher-ranked clients, which all received all 20 points in this category. (For simplicity, the Sharpe ratios given simply took the 10-day profit and loss figures divided by their standard deviation, as opposed to the more standard annualized values.)

Although the Sharpe ratio accounts for the risk-return trade-off, it's insensitive to the amount of data available, so it's generally an unreliable indicator of statistical significance. We thus also provide 95 percent confidence intervals around each client's average. By this measure, two clients (CBR-SOBI and OBVol again) had confidence intervals lying exclusively in the region of positive earnings. They 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 with high confidence that SimpleTrend is 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 clients' point totals and their profit and loss totals was 0.41; so, profitability was considerably important but not dominant. This isn't surprising, considering the motivation behind the criteria's design.

Analysis

Recall that a main motivation behind the design of PXS and the competition is to create a hybrid market in which an incoming stream of real market data can influence or "correct" a diverse market of virtual clients. We're thus naturally interested in examining the extent to which the internal PXS market correlated with or deviated from the Island market. Overall, the competition seems to have quite successfully balanced the virtual and external markets' influences. A typical plot of the PXS and Island last prices generally shows that over the entire trading day, the two prices were extremely close, with occasional small short-term deviations (we discuss some notable exceptions to this later).

More quantitatively, Figure 3a illustrates

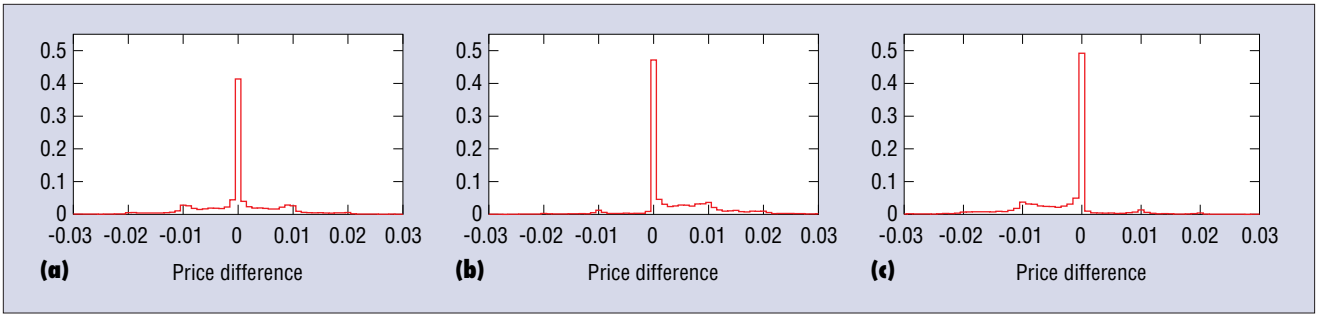

Figure 3. The differences between the PXS and Island markets for the Blue pool, aggregated over the 10 competition days: (a) last prices; (b) bids; (c) asks.

the differences between the Island and PXS last prices for the Blue pool (results for the Red pool are similar), aggregated over all 10 competition days. The greatest mass in these histograms lies close to 0 cents, and virtually all the mass is contained in a margin of ±3 cents. So, the Island data's external influence seems to have indeed caused a close, but not perfect, correlation between the two markets. The histograms' symmetry around 0 indicates that PXS deviations from the Island price are unbiased, as you might expect.

Figures 3b and 3c illustrate the differences between the PXS and Island bids and asks (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 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 PXS market.

We now examine more client-specific behaviors. Perhaps the easiest way to visualize individual clients' overall trading behavior is to examine their share positions throughout the trading day. Figure 4 contains a graph for each client in the Blue pool, with each graph containing 10 curves showing the number of shares (long or short) the client held as a function of time over the 10 competition days. The curves' crowded nature makes tracking a client's precise position on any single day difficult. However, we can infer from these curves a great deal of macroscopic information about the client's frequency and volume of trading, the bias toward long or short positions, and many other properties.

For instance, in the Blue pool (we omit detailed Red pool analysis), we see that the victorious client CBR-SOBI tended to execute relatively large transactions at relatively long intervals. This contrasts with clients such as OBMM, whose positions 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, seems incapable of anything but overall short positions. The dark mass of heavy trading activity by CReaTiv between approximately 10 and 11 am each day is probably at least partially, if not primarily, responsible for the heavy increases in overall PXS volume that occurred about that time in the Blue pool on several trading days.

The Red pool's SimpleTrend deserves special mention because its behavior nicely demonstrates that an internal pool of diverse, aggressive virtual clients can act as a proxy for the market impact that excessively large orders typically have in the real world. Unlike the other clients in either pool, SimpleTrend often engaged in sudden transactions (both buying and selling) for close to 100,000 shares, as Figure 5 shows. These large deals proved disastrous for SimpleTrend and profitable for its trading partners, because the huge orders ate deep into the opposing book and left SimpleTrend with progressively worse prices. (Recall that SimpleTrend had the worst P&L performance and actually passes a statistical significance test for unprofitability.) Despite the previously discussed statistical closeness of the Island and PXS markets, SimpleTrend's behavior caused the Red pool last price to instantaneously deviate dramatically from Island's several times. For example, Figure 5 shows where large SimpleTrend orders directly cause sudden changes in the PXS price at three distinct moments during the day. Figure 5 illustrates a typical day for SimpleTrend, where several precipitous tumbles deeper into the red are directly aligned in time with the large position changes.

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. Nonetheless, we were pleased overall with the PXS market's realism, liquidity, and tightness; the external Island data's healthy influence; the corrective effects of the other virtual clients in the case of SimpleTrend; and the client population's diversity.

The Penn-Lehman Automated Trading Project is a work in progress; we're actively planning extensions to all aspects of the project. On the systems and platform side, we're enhancing the client API, improving the speed and robustness of the execution engine's internal algorithms, and designing a Web-based GUI that will permit remote participants to use PXS and join our competitions. On the strategy side, we're 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. If you're interested, contact Michael Kearns at mkearns@cis.upenn.edu. ■

Acknowledgments

Berk Kapicioglu and Byoungjoon Kim initially developed and tested the Penn Exchange Simulator, and Narayanan Mahesh, Nick Montfort, and Rashid Tuweiq made numerous contributions to the system. The Proprietary Trading Group at Lehman Brothers in New York City gave generous scientific, professional, and financial support to the project. Michael Kearns especially thanks his Lehman colleagues Michael Bleich, Michael Bos,

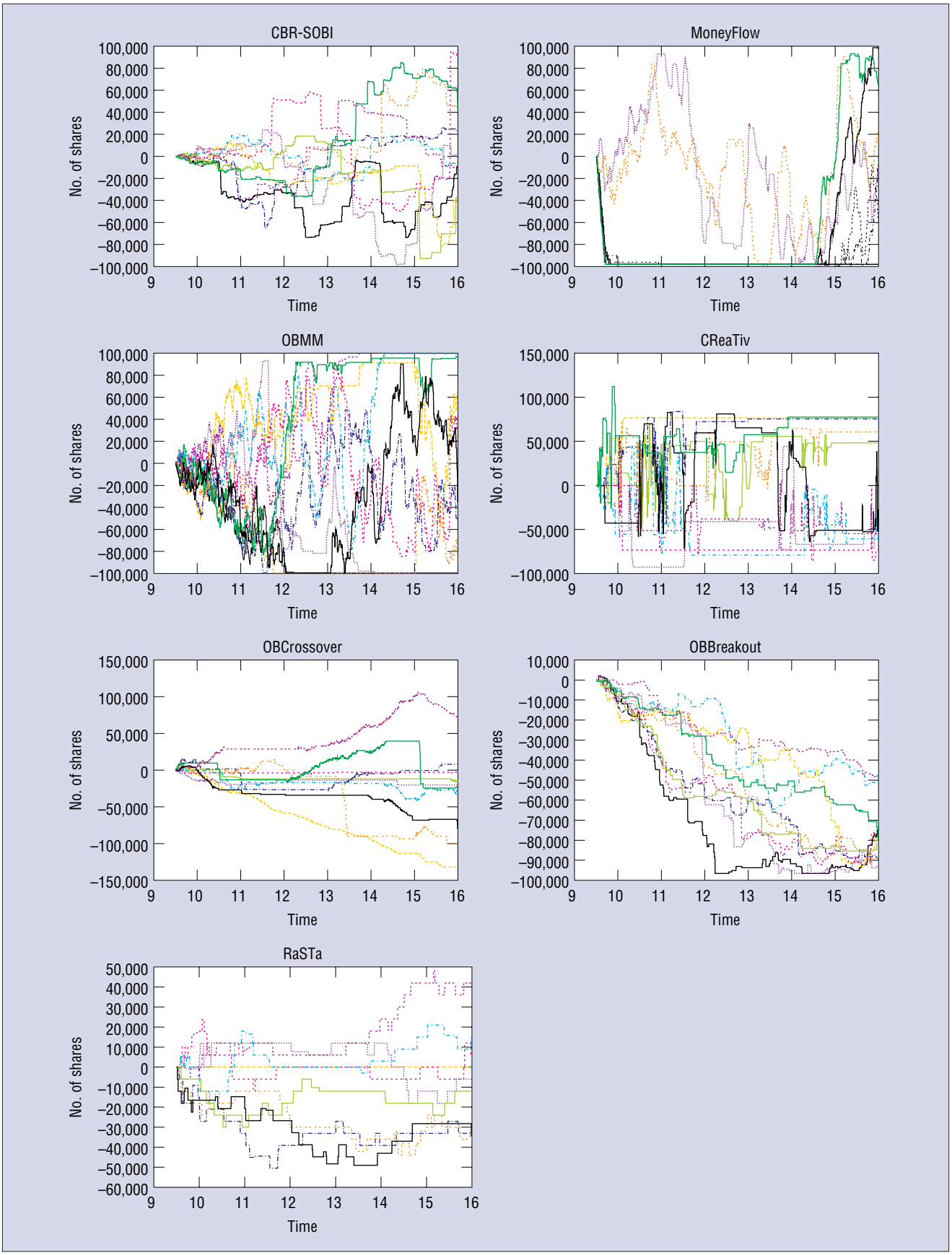

Figure 4. Blue pool client share positions over time (9:30 am to 4 pm). Each graph shows the number of shares a client owns or owes as a function of time, with each curve corresponding to a day of the competition.

The Authors

Michael Kearns is a professor of computer and information science at the University of Pennsylvania, where he has a joint appointment in the Wharton School and is codirector of the Institute for Research in Cognitive Science. His research interests include AI and machine learning, computational aspects of game theory and economics, and computational finance. Contact him at the Dept. of Computer and Information Science, Univ. of Pennsylvania, 507 Levine Hall, 3330 Walnut St., Philadelphia, PA 19104; mkearns@cis.upenn.edu; www.cis. upenn.edu/~mkearns.

Luis Ortiz is a postdoctoral researcher at the University of Pennsylvania's Department of Computer and Information Science. His research focuses on AI, computational game theory and economics, and machine learning. He previously was a postdoctoral fellow and researcher at the same university's Institute for Research in Cognitive Science. He received his PhD from Brown University's Department of Computer Science. Contact him at the Dept. of Computer & Information Science, Univ. of Pennsylvania, Levine Hall, 3330 Walnut St., Philadelphia, PA 19104-6389; leortiz@linc.cis.upenn.edu.

Andy Ellner, Amy Papandreou, Colin Rust, and Mark Sanborn for their time and knowledge. We also thank all the participants in the Penn-Lehman Automated Trading Project who developed PXS clients for their hard work and patience. We're particularly grateful for the suggestions and comments of Yuriy Nevmyvaka of Carnegie Mellon University and Peter Stone of the University of Texas at Austin.

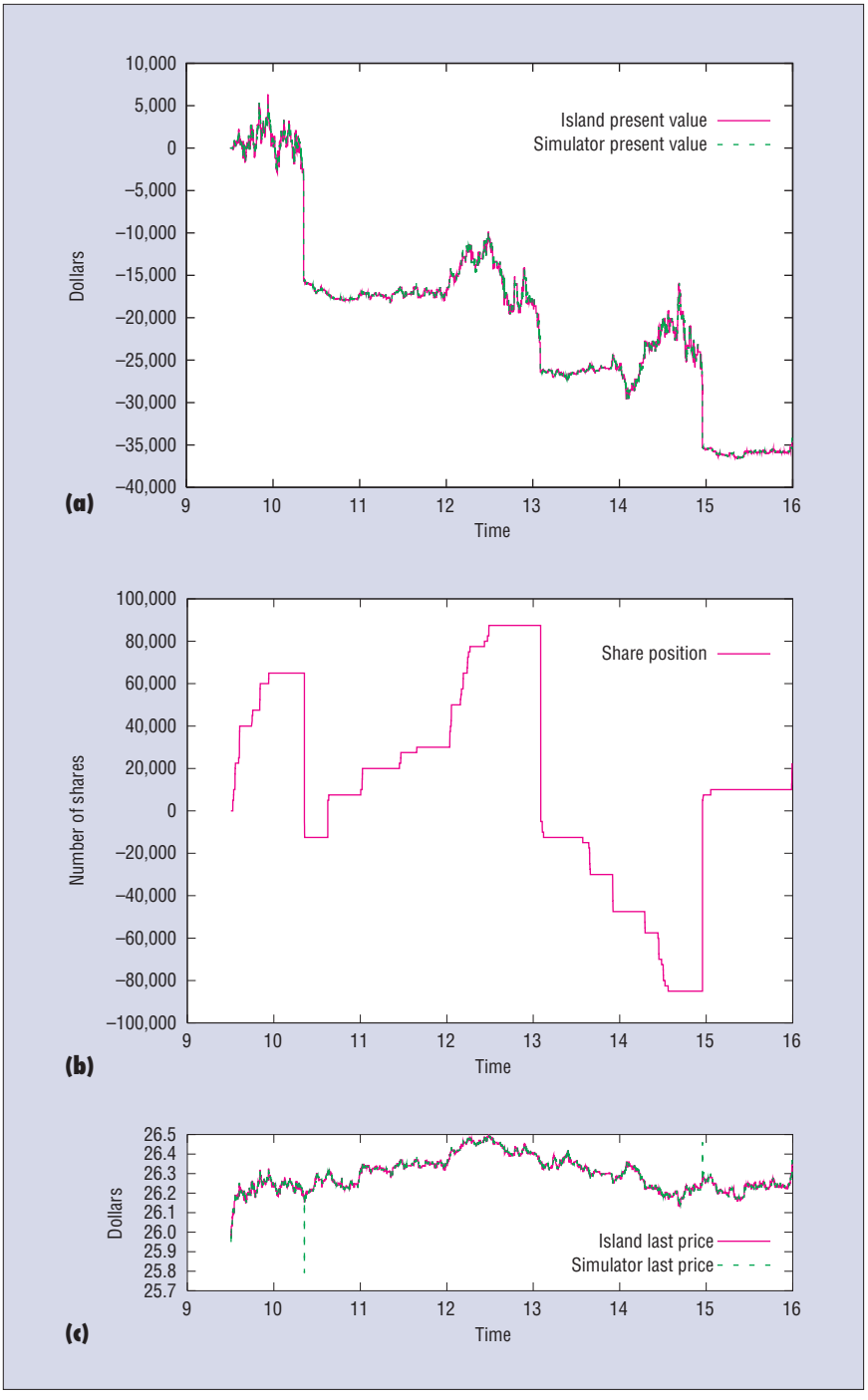

Figure 5. The client SimpleTrends' behavior for 9 May: (a) profit and loss; (b) share position; (c) share price. Three trades exceeding 60,000 shares each (a sell shortly after 10 am, a sell around 1 pm, and a buy around 3 pm) are each accompanied by sharp losses because the large orders eat deep into the PXS order books. Although the last prices are indistinguishably close for almost the entire day, the first and last of SimpleTrend's large trades cause instantaneous, large deviations of the PXS price in the corresponding direction. At the large trade around 1 pm, the PXS books apparently had enough depth near the inside market to absorb the trade and prevent a deviation between the Island and PXS prices.

References

1. *Proc. 4th ACM Conf. Electronic Commerce*, ACM Press, 2003.
2. P. Stone et al., "ATTac-2000: An Adaptive Autonomous Bidding Agent," *J. Artificial Intelligence Researc*h, vol. 15, Sept. 2001, pp. 189–206.
3. M. Wellman et al., "The 2001 Trading Agent Competition," *Electronic Market*s, vol. 13, no. 1, 2003, pp. 4–12.
4. M. Wellman et al., "Designing the Market Game for a Trading Agent Competition," *IEEE Internet Computing*, vol. 5, no. 2, Mar./Apr. 2001, pp. 43–51.
5. J. Bouchaud, M. Mezard, and M. Potters, "Statistical Properties of Stock Order Books: Empirical Results and Models," *Quantitative Finance*, vol. 2, no. 4, Aug. 2002, pp. 251–256.
6. J. Hasbrouck and G. Saar, *Limit Orders and Volatility in a Hybrid Market: The Island ECN*, working paper, Dept. of Finance, New York Univ., 2002.