AgEx: A Financial Market Simulation Tool for Software Agents

Paulo André L. De Castro[1,2], Jaime S. Sichman[2]
[1] Technological Institute of Aeronautics1, São José dos Campos, São Paulo, Brazil
pauloac@ita.br

[2]Intelligent Techniques Laboratory, University of São Paulo2, São Paulo, Brazil
jaime.sichman@poli.usp.br

Abstract. Many researchers in the software agent field use the financial domain as a test bed to develop adaptation, cooperation and learning skills of software agents. However, there are no open source financial market simulation tools available, that are able to provide a suitable environment for agents with real information about assets and order execution service. In order to address such demand, this paper proposes an open source financial market simulation tool, called AgEx. This tool allows traders launched from distinct computers to act in the same market. The communication among agents is performed through FIPA ACL and uses a market ontology created specifically to be used for trader agents. We implemented several traders using AgEx and performed many simulations using data from real markets. The achieved results allowed to test and assess comparatively trader's performance against each other in terms of risk and return. We verified that the effort to implement and test trader agents was significantly diminished by the use of AgEx. Furthermore, such results indicated new directions in trader strategy design.

Keywords: autonomous agents, software agents, autonomous asset management

1 Introduction

Many researchers have addressed the problem of creating mechanisms to automate the administration of assets. It is possible to observe the use of the most varied reasoning techniques, for instance: neural networks [1], reinforcement learning [2][3], multiagent systems [4][5][6], BDI architectures [7], Case-based reasoning [8], SWARM approaches [9] and others [10]. These initiatives could be classified by their capability of handling several assets simultaneously (multi-asset) or with just a single asset (mono-asset). It should be pointed out that the administration of several assets is more complex than just to answer the administration of an asset. It is necessary to explore the complementarities among the group of assets, especially, to minimize the portfolio risk. Therefore, it is possible to classify the cited papers in two big groups: multi-asset [4][5][6][9] and mono-asset [1][2][3][7][8][10]. We verified that there are more papers in the second group (mono-asset) than in the first group. The reason for

that becomes clear when we classify the works according with its agent goal: profit maximization [1][2][3][4][5][7][8][10] or some tradeoff solution between profit maximization and risk mitigation [6][9]. It is clear that there is a much stronger concern in maximizing profit than in mitigating risks. This balance has also happened in the beginning of the study of portfolio administration and selection. One classic paper in Finance Theory that presented a significant contribution to risk measurement and control was presented by Markowitz [11] more than fifty years ago and it helped to change the exaggerated concern with profit against risk control. This concern shows that there is a lot to advance in order to achieve software agents that may be as efficient as human beings in portfolio management. One big obstacle to research in automated portfolio management is the need for a test bed for the designed agents and systems. This test environment should be able to simulate financial markets as close to reality as possible.

This kind of tool is fundamental to research in automated portfolio management, but it is not really part of it. It is an infrastructure that could be reused by a lot of researchers. This paper presents an open source financial market simulation tool with special features that makes it different from others tools currently available. This system is called AgEx (*Agent Exchange*) and it is available under LGPL license. Such license allows its free use by researchers, even in proprietary projects.

2. AgEx Architecture

Figure 1 presents the AgEx architecture with its main components and communications links among trader agents and their human investors. The gray rectangles represent software agents (traders, manager and broker), while the circles represent the owners of the agents and a human administrator of AgEx. The entity represented by a white rectangle is a software module that is too simple to be classified as agent, and performs the actions determined by agents, such as buy and sell order executions. The component AgEx Data is just a database of real operations that took place in some real exchange and it may be used in simulations as described later.

2.1. Main Components

The AgEx system is composed by three kinds of components:

- Trader Agent: It is responsible to decide and to submit buy or sell orders to some predefined assets. In fact, these agents use the AgEx as a simulation platform framework for communication and life cycle management. Therefore, they are represented over AgEx border in figure 1. The AgEx system may provide services for many traders simultaneously, as shown in figure 1.
- AgEx Manager: This agent is responsible to validate and to process messages addressed to AgEx system. It sends the valid messages to execution that are

performed by a software module, called AgEx Broker. The execution results are received by the manager and sent to the traders that submitted the order.

- AgEx Broker: It receives and executes buy or sell orders and informs the AgEx Manager about the result of execution.

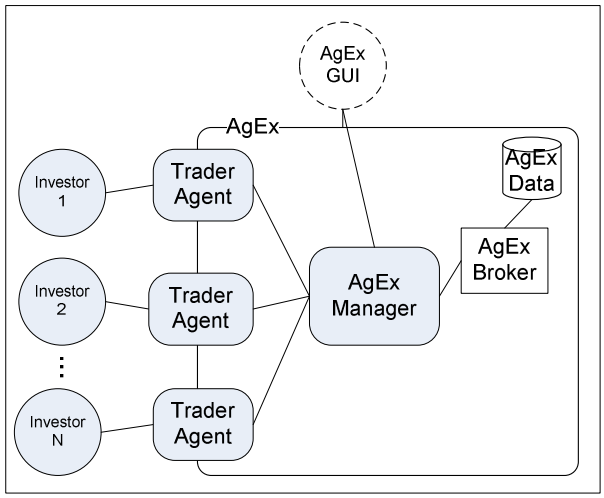

Fig. 1. AgEx Architecture and its main components and interface with external users and administrator.

2.2 Communication and AgEx Ontology

The communication within the AgEx society (manager and traders) demands a way to interchange concepts and agent actions through messages. In order to address such demand, we developed a specialized ontology to AgEx based on a content reference model created by FIPA [12]. This ontology includes the main concepts, predicates and possible actions needed by trader agents, such as: asset, quote request, quote result, order submission, order result, etc. These concepts, predicates and actions are used to create content for any message exchanged within an AgEx society. The possible concepts in AgEx are *Order*, *MarketOrder*, *LimitedOrder*, *OrderResult*, *Query*, *QueryResult*, *AssetConcept*, *Error* and *Terminate*.

Each message in AgEx has a content entity extended from *ContentElement* class. This entity may be an action that can be performed by agent (derived from *AgentAction*). In this case, the message content is a request to the agent to perform one specific action. It is interesting to notice that an agent action is also a concept. The possible agent requests in AgEx are: *Order*, *MarketOrder*, *LimitedOrder* (traders submit orders for execution), *Query* (trader request information about assets) and Terminate (the manager tells the trader that the simulation is over). The AgEx Manager uses the derived classes of Result to inform the trader about the request results. When one trader submits an order execution request (*Order*, *MarketOrder* or *LimitedOrder*), it is responded with an *OrderResult* concept. Whether one trader submits a *Query* request, it is responded with a *QueryResult* concept. Whenever an

invalid or unknown message is received by AgEx manager, it sends a message which content is an Error concept.

The communication among agents in AgEx is performed using the addressing and delivering services provided by the Java Agent Development Environment, JADE [13]. Such message service is compatible with the standards defined by FIPA [12]. The possible dialogs between the *AgExManager* and any trader agent are related to three specific situations, which are described next:

- Order submission: A trader agent decides to submit an order (buy or sell, limited or market). It creates a message with content equals to such order and sends it to AgExManager. This one returns a message with result of order execution.
- Query about some asset: The trader asks quote information about a specific asset. The AgExManager responds the message with the required information or an error message if the information is not available.
- Unknown or Invalid Messages: In case of the sent by the trader, the AgExManager responds with a message, which content is an Error concept.

2.3. Simulation Mechanism

In its default mode (historical or real price mode), AgEx allows simulation using asset information from real stock markets. This information is composed by assets prices (open, high, down and close prices) and volume (shares traded by assets). Therefore, the asset prices do not change according to trader orders: prices are defined by AgEx Data. This kind of simulation is particularly useful when the research is focused on the development of trading algorithms that do not account the effect caused by the own algorithm. In fact, this effect may be despised since the amount of assets traded by the agent is much smaller than the market volume.

However, researchers interested in analyzing the effect of some trader strategy in the market may use the second simulation mode in AgEx. It is called live price mode (or price formation mode). In live mode, the prices and volume are defined exclusively by the orders submitted from the trader agents.

The trader agents and the AgEx manager agent are synchronized by message exchanges. The manager defines the duration of each cycle (time step) and their transitions. All traders must be able to get the needed information, deliberate and submit orders within the interval of one time step. Whenever a trader doesn't complete these jobs within one time step, the system raises an overrun exception. One trader agent doesn't known in advance at which price one market order will be executed, just like it happens in real markets. Furthermore, agents are not allowed to access price information beyond the current cycle. These features provide more realism to the simulation and avoid that one trader gets privileged information.

2.4. Real Operation Mechanism

Despite the fact the AgEx is mainly concerned with simulation of financial markets, we designed it to be able to be used as a platform for software traders operation in real

exchanges. This will be possible through the replacement of the AgEx Broker by another component that adapts the interface expected by the AgExManager to the interface provided by the target market exchange. This kind of component is called AgEx-Exchange Adapter. The development of an Adapter is not complex; however it requires an express permission from a real exchange or broker company in order to access the system. We intend to implement an example of an AgEx Adapter in the future.

3. AgEx Implementation

AgEx was implemented in Java using JADE platform and it is composed by more than 10.000 lines of code shared among 101 classes and interfaces. However, the development of a new trader agent requires the creation of only one class which must extend *AgExTraderAgent* class and implement one method responsible for order definition at each cycle. The other tasks demanded as get quote information, send orders to the market, compute results are performed by AgEx itself and may be configured through a specific configuration file.

In order to make easier the launcher of an agent society, AgEx provides a launcher that allows the definition of a society through XML files including parameters for traders (initial capital and traded assets, for instance). In figure 2, an example of such a society definition file is presented.

```
  <?xml version="1.0" encoding="UTF-8"?>
<society-agex>
  <manager  remote_manager="yes" hostname="127.0.0.1"
  port="1099"/>
 <trader name="RSI" initial_capital="1000000"
 path="agex.traders.RSI">
        <asset id="AAPL" initial_shares="0"/>
        <asset id="ADBE" initial_shares="0"/>
        <param  value="para1" />  </trader>
 <trader name="MA" initial_capital="1000000"
 path="agex.traders.MA">
  <asset id="AAPL" initial_shares="0"/> </trader>
</society-agex>
```

Fig. 2. An Example of society definition files with two trader agents. The manager tag defines that this society will be connected to an AgEx server running in the host indicated by its IP number and TCP port.

After launching a society, it is possible to follow its simulation progress or to pause it using a simple graphical user interface (figure 3). Furthermore, AgEx allows launching several societies from different computers at different times to the same market simulation (see figure 4). These societies are synchronized by the manager in order to observe the same simulation time.

In figure 4, we present an example of two societies launched from different computer in JADE Management GUI (in JADE, each container is associated to one computer), that are associated to AgEx manager in a third computer (Main-container). In the first container (Container-1), there are two traders agents (RSI and MA), while the second container has three agents (PriOsc, Sthocastic and MACD). All five trader

agents deal with the agexManager agent located at the Main-container. The strategies used to build such trader agents are described later.

Fig. 3. AgEx Manager GUI

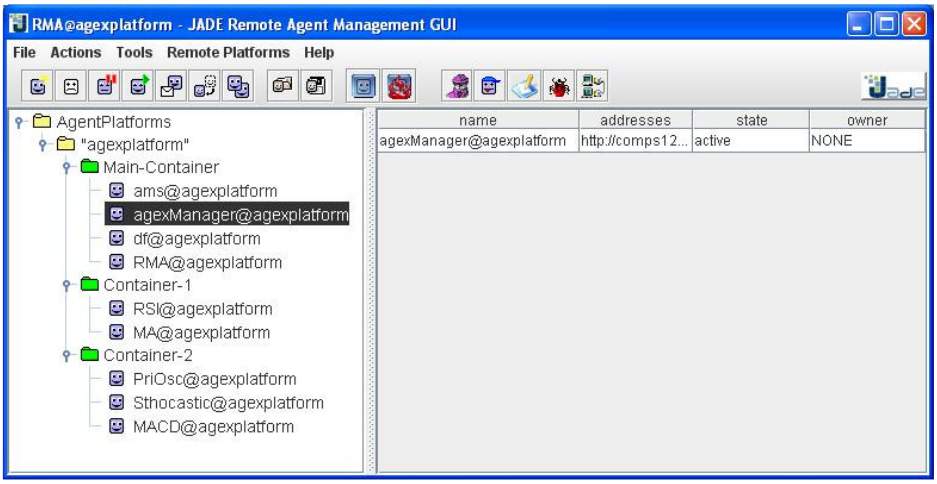

Fig. 4. AgEx trader agents distributed in three computers

3.1. Simulation Generated Data

AgEx registers in the end of each cycle the position of each trader (money, shares, stock prices and orders). Furthermore, it creates a summarized file with the results of all traders in the computer that runs the Manager. These files are created in csv format that facilitates their analysis with spreadsheet programs (like Excel or Open Office).

3.2. Importing Data

Real quote information is essential to perform simulation of markets and also to provide data to agents that trade in real markets. Fortunately, several web sites (like Yahoo Finance, for instance) provide this kind of information free of charge. This information must be inserted into AgEx Data to be used by the system. AgEx Data is implemented as a Firebird RDBMS. We created a GUI to import quote information

files with Yahoo Finance format. It makes easier the capture and information update in AgEx tool.

4. Related Work

In this section, we present a comparative analysis of some selected systems with similar propose of AgEx. Such analysis is based on some features that allow or facilitate the simulation of markets to test and assess trader agents. We do not intend to judge the overall quality of the cited systems, but just identify differences (positive and negative) with the system proposed here.

The selected systems for analysis are eAuctionHouse [14], eMediator [15], PXS [8], SFI [16] and JASA [17]. In table 1, we present a comparative analysis of these systems based on four features.

The two first features (real and live price modes) were already discussed in the Simulation Mechanism section. The third feature indicates if the system source code is available free of charge. The fourth feature tells whether the system defines or uses ontology to exchange information (concepts or requests). AgEx is the only one that fulfills all features. In fact, it is the only open source tool able to perform historical price simulations. Additionally, we may say that AgEx is the only system that is adherent to FIPA recommendation for communication among agents. Despite the fact some may say that this is not clearly an advantage, we may argue that the adherence to standard communication patterns makes easier its use by others researchers.

Table 1 – Comparison among Selected Systems

System	**eAuctionHouse**	**eMediator**	**PXS**	**SFI**	**JASA**	**AgEx**
Real Price mode	No	No	Yes	No	No	Yes
Live Price Mode	Yes	Yes	Yes	Yes	Yes	Yes
Open Source	No	No	No	Yes	Yes	Yes
Use ontology	No	No	No	No	No	Yes

5. AgEx in Action

We used AgEx platform in many simulations (sections 5.2, 5.3 and 5.4) in order to analyze five trader strategies based on technical indicators: *RSI*, *Price Oscillator (PriOsc)*, *Moving Average (MA), Moving Average Convergence-Divergence (MACD)* and *Stochastic*. Such indicators are widely used by financial analysts as part of their decision process. These strategies were implemented as single trader agents and each one took less than 150 lines of Java code: this indicates that AgEx really reduces the implementation effort of trader agents. We are not going to detail such strategies in this paper because they are explained in detail in the indicated references [6][18].

In order to assess the trader's performance against the performance of their assets, we developed another trader agent that simply buy and hold one unit of each asset that it manages. This agent (*BuyAndHold*) is useful to give information about the evolution of asset prices. It's expected that a good trading strategy overcome the buy and hold strategy.

5.1. Experimental Setup

Frequently, studies on automated asset management present very limited experimental evaluation, for instance using one or very few assets and/or short evaluation periods. Another concern should be to avoid unclear selection criteria of assets and periods to avoid bias in such selection. These problems may cause wrong conclusions and dangerous generalizations about the agent's performance in periods and assets that were not taken into account.

We tried to avoid this problem by selecting a long period (19 years) and several assets: stocks of 14 companies from 5 different economic sectors (technology, healthcare, services, consumer goods and apparel stores). We selected companies from Nasdaq 100 Index, which lists the 100 more relevant companies on Nasdaq Exchange.

Unfortunately, many companies listed are relatively new and therefore they don't present long temporal price series. In fact, there are only than 10 companies with at least 20 years of history. We preferred to reduce one year in the period, in order to get 14 assets with available price series of 19 years. These assets and companies are presented on table 2.

Table 2 - Selected Stocks

ID	Name	ID	Name
AAPL	Apple Inc	DELL	Dell Inc
ADBE	Adobe Sys. Inc	INTC	Intel Corporation
ALTR	Altera Corp	JAVA	Sun Microsystems
AMAT	Applied Materials Inc.	MSFT	Microsoft Corp.
AMGN	Amgen Inc	ORCL	Oracle Corp.
CMCSA	Comcast Corp	PCAR	PACCAR Inc.
COST	Costco Wholesale Corp.	ROST	Ross Stores Inc.

5.2. Risk and Return Performance

We have performed simulations of six trader agents (*RSI, MACD, MA, PriOsc, Stochastic, BuyAndHold*) over the period of Jan 1, 1989 until Dec 31, 2007, where each agent were able to trade with 14 stocks (listed on table 2). The obtained results are presented in terms of annual return and risk (measured as standard deviation of agent's patrimony), in figure 5 and 6, respectively. These results show that there is no trader that overcomes the others in a consistent way over the whole period. In fact, several traders are replacing each other in the position of best performance as time evolves. This is true also when analyzing the traders under risk criteria. Tables 3 and 4 present these results clearer. In terms of final return, *RSI* obtained the best performance in six years and the second best in two years. However, we may see that *MACD* has very similar overall performance, because it achieved the best performance in others five years, the second best performance in two years and furthermore it got the third best in four years against only one of *RSI* trader. The others traders presented inferior results, but all got the best performance in some year

except the *Buy and hold* trader. Therefore, we may conclude that there is no strong superiority of any analyzed traders regarding the final return. Probably, one trader strategy that uses a mix of the analyzed strategy could get better results. Furthermore, the poor performance of *Buy and Hold* trader makes it clear that is possible to achieve good results using active strategies.

Table 3. Final return results achieved by traders. Traders are sorted in alphabetical order. The ranking is defined by the number of times the trader achieved first, second or third places.

Trader	***Ranking***	***1o.***	***2o.***	***3o.***
Buy And Hold	4	3	6	6
MA	2	5	2	1
MACD	5	1	3	4
PriOsc	1	6	1	1
RSI	6	0	5	5
Sthocastic	3	4	2	2
Total	**-**	**19**	**19**	**19**

In table 4, we may realize that traders with good performance according return criteria achieved this result at cost of higher risk. One may observe that RSI (first in return) has become the last in risk evaluation and MACD (the second in return) was just third in risk evaluation. Moreover, Buy and Hold trader (the sixth in return) is the second in risk evaluation. This performance inversion is not a surprise. In fact, it is compatible with the common notion that in order to achieve higher returns, it is necessary to accept higher risks.

Table 4. Final risk results achieved by traders.

Trader	**Ranking**	**1o.**	**2o.**	**3o.**
Buy And Hold	2	4	6	5
MA	5	2	7	2
MACD	3	4	0	4
PriOsc	6	0	3	2
RSI	1	6	2	1
Sthocastic	4	3	1	5
Total	**-**	**19**	**19**	**19**

5.3. Broker's Fee Influence

One common assumption in autonomous traders design is that as fees will be charged from all traders no matter its strategy, then strategies could be designed and compared among themselves without concern about fees, because they would reduce profitability of all traders in an almost equal way. The AgEx supports fees collection (a fixed amount by operation and/or a percentage of transaction volume). Therefore, we used this feature to verify this common assumption. We repeated the scenario described in section 5.2, but charging 10$ dollars and 0.5% of order volume (shares times price) per each order. Despite the fact, fees change very much among different brokers, these values may be considered typical. The summarized results obtained by trader agents for the same period and asset set used in section 5.2 is presented in table 5. In fact, achieved results showed that there is profitability reduction, but some

agents were more affected than others. Observing table 5, one may realize that *Buy And Hold* agent is in better position than in table 3. This happened because this agent submits fewer orders than others so it paid fewer fees. One trader may benefit from submitting fewer orders, but each order with higher volume.

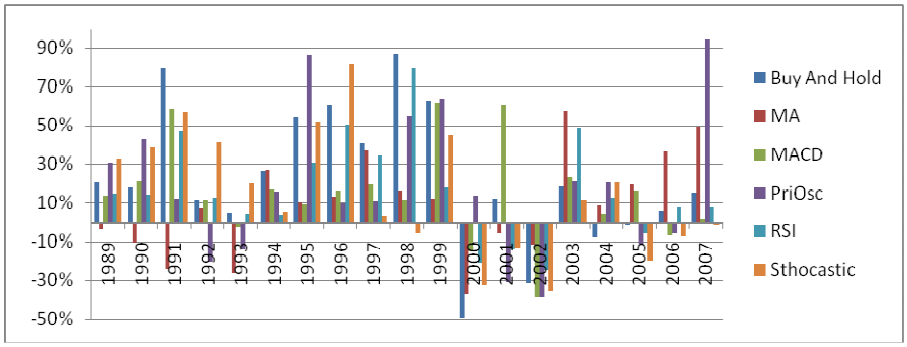

Fig. 5. Trader Agent's Return by year.

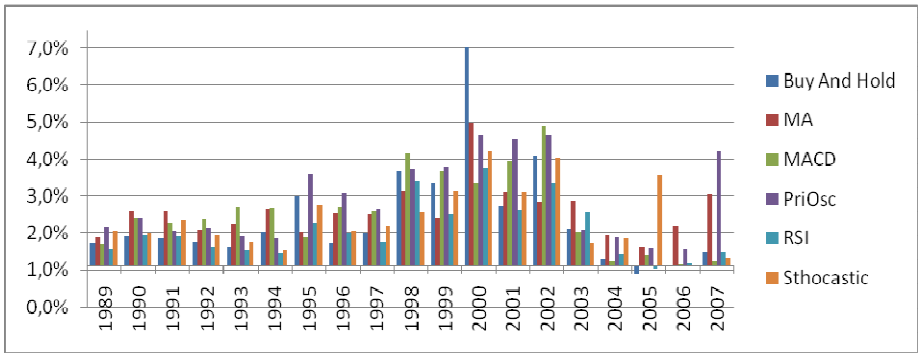

Fig. 6. Trader Agent's Risk. The risk is assessed as the standard deviation of agent's returns.

5.4. Trader Performance by Paper

Table 5. Final return results achieved by traders in simulation with fees.

Trader	***Ranking***	***1o.***	***2o.***	***3o.***
Buy And Hold	2	4	6	5
MA	5	2	7	2
MACD	3	4	0	4
PriOsc	6	0	3	2
RSI	1	6	2	1
Sthocastic	4	3	1	5
Total	**-**	**19**	**19**	**19**

Figure 7 presents daily average performance achieved by the trader agents for each paper. These results are obtained through simulation over the period from 2003 to 2007; each trader was allowed to deal with one asset. We realized that one agent with

very good performance for an asset may get very poor results in another. For instance, the *RSI* agent was the first for AAPL and in the same period the fourth for AMGN.

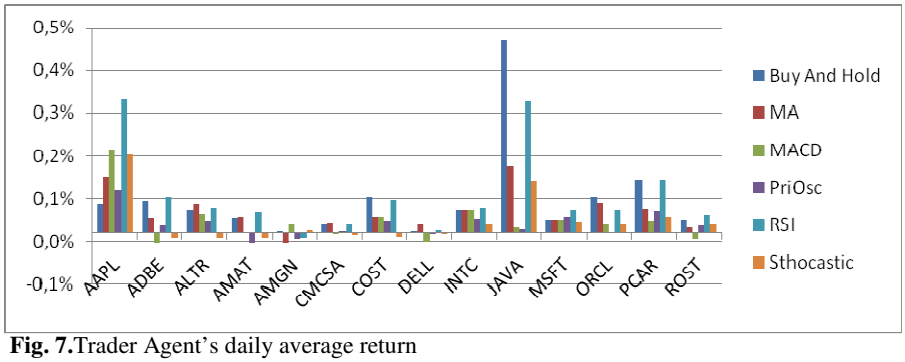

Fig. 7. Trader Agent's daily average return

6. Conclusions

The AgEx tool presented in this paper is a special-purpose software agent platform for simulation of financial markets. It is open source and allows market simulation with prices from real markets. It makes available a market ontology that simplifies communication. AgEx provides facilities to launch traders from several computers over the net and to analyze their performances. We have presented six trader agents implemented using AgEx and the obtained results from their simulation in several scenarios. In these implementations, we could realize that the effort to implement trader agents was significantly reduced by AgEx use. Furthermore, AgEx is adherent to international standards of agent communication [12]. This feature may facilitate its use by others researchers.

We have performed a significant amount of simulated experiments (over a period of 19 years, using 14 different assets) and tested the influence of broker's fee in trader performance. The obtained results for trader's performance were analyzed in terms of risk and return. The comparison among traders dealing with and without fees showed that the presence of fee may harm less one agent than others (section 5.3). The results also showed that there is no dominant strategy along the time (section 5.2) and no agent presented best performance for all papers (section 5.4) among analyzed traders.

Moreover, these analyses make us believe that new strategies mixing information from existing traders may achieve good results. We intend to use AgEx in our future research to develop this kind of trading strategy. Finally, we believe that AgEx can be very useful for others researchers trying to develop new strategies for automated asset management.

Acknowledgments. Jaime Sichman is partially supported by CNPq/Brazil.

References

1. Kendall, G.; Su, Y. "Co-evolution of successful trading strategies in a simulated stock market." In: proceedings of ICMLA'03. Los Angeles: [s.n.], p. 200–206. (2003)

2. Sherstov, A.; Stone, P. "Three automated stock-trading agents: A comparative study." In: Proceedings of AMEC Workshop - AAMAS 2004. New York: (2004)

3. Nevmyvaka, Y.F.Y.; Kearns, M. “Reinforcement learning for optimized trade execution.” In: Proceedings of the 23rd International Conference on Machine Learning- ICML 2006. Pittsburgh, Pennsylvania: [s.n.], (2006)

4. Decker, K.; Pannu, A.; Sycara, K.; Williamson, M. “Designing behaviors for information agents.” In: JOHNSON, W. L.; HAYES-ROTH, B. (Ed.). Proceedings of Agents’97. New York: ACM Press, p. 404–412. (1997)

5. Luo, K.L.Y.; Davis, D.N. “A multi-agent decision support system for stock trading.” IEEE Network, v. 16, n. 1, p. 20–27, (2002)

6. Castro, P. A.; Sichman J. S. “Towards cooperation among competitive trader agents.” In: Proceedings of 9th ICEIS. Software Agents and Internet Computing track. Funchal, Madeira - Portugal: [s.n.],. pp.138 – 143. (2007)

7. Feng, X.; Jo, C.H. Agent-based stock trading. In: proc. of the ISCA CATA-2003. Honolulu, Hawaii: [s.n.], (2003)

8. Kearns, M.; Ortiz, L. “The penn-lehman automated trading project.” IEEE Intelligent System, v. 18, n. 6, p. 22–31, 11-12 (2003)

9. Kendall, G.; Su, Y. “A particle swarm optimisation approach in the construction of optimal risky portfolios.” In: proc. of the 23rd IASTED. Innsbruck, Austria: [s.n.],. p. 140–145. (2005)

10. Feng, R.Y.Y.; Stone, P. Two stock-trading agents: Market making and technical analysis. In: Proceedings of the Agent Mediated Electronic Commerce (AMEC) Workshop - AAMAS 2003. Melbourne, Australia: (2003)

11. Markowitz, H. M. “Portfolio selection.” Journal of Finance, v. 7, n. 1, pp. 77–91, (1952)

12. FIPA. The Foundation for Intelligent Physical Agents. Website: <www.fipa.org>

13. Bellifemine, F.L.; Caire, G. and Greendwood, D. “Developing Multi-Agent Systems with JADE” Ed. Wiley. (Wiley Series in Agent Technology) April (2007)

14. Wurman, P.R. , Wellman, M.P. e Walsh, W.E. “The Michigan Internet AuctionBot: A configurable auction server for human and software agent”, AGENTS, pp. 301-308, Minneapolis/St. Paul, MN, (1998)

15. Sandholm, T; “eMediator: A Next Generation Electronic Commerce Server”; International Conference on Autonomous Agents (AGENTS), Barcelona, June (2000)

16. LeBaron, Blake, "Building the Santa Fe Artificial Stock Market",Working Paper. Brandeis Univ. (2002)

17. Phelps, S.. “Evolutionary Mechanism Design.”. Phd Thesis. Univ. of Liverpool. (2007)

18. Market Screen Investment Tools website. URL: . <http://www.marketscreen.com>