COAST: An architecture based on negotiation among competitive agents for automated asset management

Paulo André Lima de Castro[1], Jaime Simão Sichman[2]
(1) Technological Institute of Aeronautics - ITA
São José dos Campos , SP, Brazil
pauloac@ita.br
(2) Intelligent Techniques Laboratory, University of São Paulo
São Paulo, SP, Brazil
jaime.sichman@poli.usp.br

Abstract

*In order to manage their portfolios in stock markets, often human traders use a set of algorithms and/or indicators, which are based on stock prices series. These algorithms are usually referred to as technical analysis. However, traders prefer to use a combination of various algorithms, rather than choosing a single one: the several signals provided by these algorithms and their own knowledge are combined to determine the orders to buy or sell some stocks. Inspired by the human traders´ decision processes, our architectural approach composes heterogeneous autonomous trader agents in a competitive multiagent system. This architecture allows the use of various algorithms, based on different technical analysis indexes to manage portfolios. This architecture is named COAST (**CO**mpetitive **A**gent **S**ocie**T**y) and it is composed by two kinds of agents:* advisors *that analyze the market situation autonomously and competing with each other for resources; and* coaches *that are able to coordinate several advisors and negotiate with each other to define the best money allocation within the society. This negotiation is performed using a negotiation protocol proposed in this work. We have implemented a system using COAST architecture, using a financial market simulator called AgEx. This system was tested using real data from the Brazilian stock exchange. The test results have shown a good performance when compared to an adjusted market index. The negotiation protocol used by coach agents provided a mechanism to easily integrate new trading algorithms, without the notion of a central agent or a centralized decision mechanism, which is a highly desirable feature in scalable multiagent systems.*

1 Introduction

Multiagent systems have been used in many real problems, such as business management workflow, information management, electronic commerce, air traffic control and social simulation among others (Wooldridge, 2002). Multiagent approaches are especially interesting in problems that are naturally distributed, complex and dynamic where autonomous entities (agents) can handle some aspect of the problem and they act in a cooperative way to achieve common goals.

In this paper, we intend to use a multiagent approach in automated asset management, which is a particular case of these distributed, complex and dynamic environments. We propose here a multiagent architecture, COAST (**CO**mpetitive **A**gent **S**ocie**T**y), which is based on *competitive agents* that act autonomously on behalf of an investor in financial asset management. There are two classes of agents in the architecture. The first one composed of *competitive* agents and the second one composed of *partially cooperative* agents. Each one of the agents of the first class, called *advisors*, is able to manage one single asset according to one particular trading strategy defined a priori. The goal of advisor is to keep the highest possible evaluation from the point of view of his coach. Despite the fact that agents make their own decision independently of others, the control that one keeps over society's resources is limited and shared. Agents that belong to the second class, called *coaches*, are responsible to manage one single asset and hence are in charge of evaluating their advisors performance. The goal of each coach is to maximize the return of the investor's portfolio. Such evaluation is used in a resource distribution process, which is negotiated among the coaches. COAST architecture tries to allocate more resources to the agents with better performance.

The rest of the paper is organized as follows. In section 2, the main concepts related to automated asset management are presented, as well as previous multiagent approaches in this domain. The main contributions of the work, the COAST architecture is described in section 3. The experiments that we have performed with the architecture are described and analyzed in section 4. Finally, we present in section 5 our conclusions and further work.

2 Automated Asset Management

The ultimate goal of an asset manager, automated or not, is to find out and adopt the most desirable set of assets for an investor, according to his preferences. The manager may adopt one set of assets through the *submission of buy and sell orders* to the stock market. The buy and sell transactions and price formation are defined through the processing of the orders of all investors in the market. In this section, we briefly present two types of analyses used in asset management: technical and fundamentalist analyses (section 2.1) and describe some related work in the automated asset management domain (section 2.2).

2.1 Technical and Fundamentalist Analyses

In the asset management domain, there are many analytic strategies based on time series analysis, which are often grouped in an approach called *technical analysis*. These strategies use some market information to identify patterns and to define orders. Some examples are *moving average*, *moving average converge-divergence*, *stochastic* and *relative strength index (RSI)*, but there are many others strategies (Castro and Sichman, 2009).

Another approach to trading strategies is called *fundamentalist analysis*. It is based on information related to economic fundamentals (including company, sector and macroeconomic fundamentals), such as net profit, market share, revenues, sector growing rates, global growing rate among others. The fundamentalist analysis approach is less used in automated asset management, despite the fact it is widely used by human asset managers. This choice is due to the greater complexity to represent in an algorithm many fundamentalist concepts, while it is quite easier to design algorithms to calculate time series used in technical analysis (Araújo and Castro, 2010).

Even within technical analysis, the identification of which information is really used and how the deliberation process occurs may change dramatically among different strategies. Furthermore, strategies may present very different performance according to market scenario (Castro and Sichman, 2007). This observation brought one first guideline to our architecture, i.e., to facilitate the composition of different strategies, like shown in section 3.

2.2 Related Work

Automated asset management, also known as automated stock trading, algorithm trading, high frequency trading and some others terms, has been a focus for many researchers (Decker et al., 1997; Yuan Luo and Davis, 2002; Kearns and Ortiz, 2003; Sherstov and Stone, 2004; Kendall and Su, 2003; Feng and Jo, 2003). It is possible to identify two different groups, according to the typical time interval between orders (or position-holding period). Strategies that have to deal with short time intervals, like weeks, few days or even fractions of second cannot be based on fundamentalist analysis, because this latter is focused on long period scenario and it is can be used only when the typical time between orders are months or years. Therefore, technical analysis is widely used for short time intervals. When the holding-position period is very short, less than one day, it is often called high frequency trading. Many researchers and practitioners have been developing algorithms to achieve better performance exploring the fact that an automated system can analyze a significant higher amount of information when compared to a human being in small time periods (Aldridge, 2009; Durbin, 2010). As far as we know, few initiatives try to explore complementarity among trading algorithms (one exception is (Castro and Sichman, 2007)), and as long we know none try to explore complementarity through a negotiated process, as it is proposed in this paper.

3 COAST Architecture

The COAST architecture is designed to facilitate the simultaneous use of many trading strategies and to explore the competition among these strategies, in order to achieve better results to the whole society. These strategies are materialized through agents called *advisors*. In COAST, strategies outputs are not interpreted as orders, but as advices about one specific asset. The other architectural guidelines are the following: **(i)** it should work with many different assets, **(ii)** it should adapt strategies' relevance for each asset and **(iii)** it should avoid central agents or a centralized decision making procedure about resource allocation through assets. In fact, there are multiple coordinator agents, called *coaches*, that negotiate among themselves and try to achieve the best possible performance to the investor. Each coach is specialized in one single specific asset. Therefore, a society with four assets and three different strategies would be composed by four coaches and twelve advisors (three advisors for each coach), like shown in figure 1. The advisors located in a same column operate with the same asset

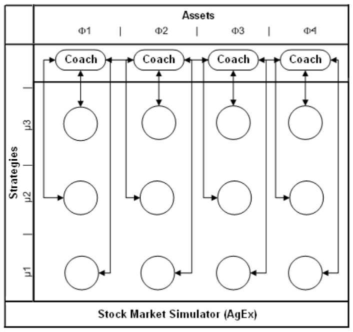

Figure 1. Example of a COmpetitive Agent SocieTy that manages four assets composing three strategies.

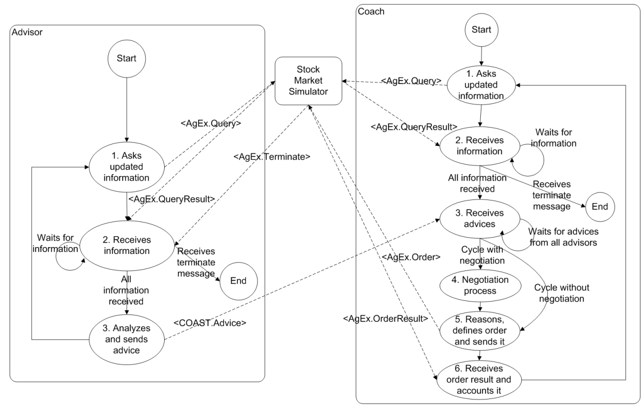

Figure 2. Advisor and coach life cycles in COAST architecture.

and the coach in top of the column evaluates and coordinates the work of the advisors in the column. According to these evaluations, one advisor with good performance has more relevance in coach decisions than the other advisors. Coaches auto-evaluate and negotiate among themselves to allocate more money to the coaches with better performance in the society, as described in section 3.3. It is important to notice that we model the architecture considering **autonomous** agents acting as experts for a specific asset, namely the coaches. Therefore, there is no central agent that controls the other agents. Indeed, coaches need to negotiate to solve conflicts and to work together.

3.1 Advisors

Advisors suggest to buy or to sell a number of shares of a specific asset following their own strategy and their goal is to give the best possible advices to improve portfolio return. Advisors can be easily created using any well known trading strategy. These advices are sent to the coach, who is the agent in charge of order definition. The advisor's life cycle is presented on the left of figure 2. In this figure, dashed lines show messages exchanged between agents and solid lines show state changes for each agent. Each state is represented by an ellipse, and has the following meaning :

1. **Asks for updated information**: The advisor, according to its strategy, asks for updated information from the *stock market simulator* (Castro and Sichman, 2009), which can be seen in the center of figure 2.
2. **Receives information**: The stock market simulator returns the information which is locally stored. This step is also used to synchronize all the agents, in simulated time.
3. **Analyses and sends advice**: According to the collected information and his strategy, the advisor defines and sends a buy/sell/hold advice to his coach.

The advisors performances are evaluated by their coach according to their advices and the market evolution. For instance, whenever an advisor suggests buying an asset whose price arises after the advice, this advisor is positively evaluated. A similar reasoning can be made regarding a selling advice.

3.2 Coaches

Coaches basically *receive* advices, *evaluate* their advisors, *negotiate* with others coaches and *define orders* that are submitted to the market. These activities are performed along all the agent lifecycle. However, the negotiation process does not happen in all cycles, only at periodic intervals which include several cycles. Negotiation in all cycles would be senseless, since the previous negotiated allocation would not have had enough time to be tested. This negotiation period is one of COAST society parameters. The coach activities are presented on the right of figure 2, and have the following meaning:

1. **Asks for updated information**: The coach asks for new information about its target asset. Even if coach orders are completely based on the messages sent by its advisors, he would need asset information to evaluate them.
2. **Receives information**: The *stock market simulator* (Castro and Sichman, 2009) returns the stored information to be used on advisors' evaluation.
3. **Receives advices**: The coach receives advices from all advisors that deal with the same particular asset that he manages. This activity is kept until he receives advice from all his advisors. During this activity, he also

R1. **If** *Advice* **is** *Buy* **and** *Evaluation* **is** *High*
Then *Expectation* **is** *Strong bullish*
R2. **If** *Advice* **is** *Sell* **and** *Evaluation* **is** *High*
Then *Expectation* **is** *Strong bearish*
R3. **If** *Advice* **is** *Buy* **and** *Evaluation* **is** *Medium*
Then *Expectation* **is** *Bullish*
R4. **If** *Advice* **is** *Sell* **and** *Evaluation* **is** *Medium*
Then *Expectation* **is** *Bearish*
R5. **If** *Advice* **is** *Manter* **Then** *Expectation* **is** *Unbiased*
R6. **If** *Evaluation* **is** *Low* **Then** *Expectation* **is** *Unbiased*

Figure 3. Coach expectation definition rules.

performs the advisors evaluation, as described in section 3.1. Whether the current cycle must include negotiation, next step is activity 4, otherwise next step is activity 5.

4. **Negotiation Process**: The coach performs negotiation with others coaches in order to define a new resource (money) allocation. This activity is much more complex than the others and therefore is described separately in more detail in section 3.3.

5. **Reasons, defines orders and sends them**: Based on advisors suggestions and following a set of fuzzy rules, the coach defines its order and sends it for execution to the *stock market simulator* (Castro and Sichman, 2009).

6. **Receives order result and accounts it**: The coach receives the order result, whose value may be total, partial or not executed at all, and accounts it in his portfolio, including the real price that was used to buy or sell the asset.

Coaches also calculate their expectation about their own performance in the near future. This performance **expectation** is modeled as a linguistic variable with five terms: *strong bearish*, *bearish*, *unbiased*, *bullish* and *strong bullish* (Pedrycz and Gomide, 1998). The coach expectation formation is based on the information (advice) that comes from his advisors and their respective evaluation. The **advice** is also a linguistic variable, with three linguistic terms: *sell*, *hold*, and *buy*. Finally, advisor **evaluation** is also a linguistic variable and have three linguistic terms: *low*, *medium* and *high*, where the universe of discourse is the success rate of the advisor [0%-100%]. Each coach receives advices from its advisors and follows the fuzzy rules presented in figure 3, in order to define its expectation.

The coach expectation is used in **order definition** and also in the negotiation mechanism when a coach must decide whether he should transfer some of his current resources to a coach with better performance. As described in section 3.3. The coach defines his order according to his expectation using a fuzzy decoding method (in our implementation, center of gravity method). For instance, a *strong bullish* market leads to a buy order with high volume, meanwhile a *strong bearish* expectation leads to a sell order with high volume and a *unbiased* expectation makes the coach to keep its current position.

3.3 Negotiation Mechanism

A negotiation mechanism is defined by a *negotiation protocol*, composed by the *communication rules* among participants, and by the players *strategies* (Rosenschein and Zlotkin, 1994). Coaches have individual and social preferences and negotiate guided by those preferences. In this section, we describe first these individual and social preferences, then the proposed negotiation protocol and finally the roles that a coach can assume in each negotiation round.

3.3.1 Individual and Social Preferences

Coaches are partially cooperative because they have a global social goal to achieve: maximize the return of society portfolio, but they try to overcome the others and get more resources to themselves, therefore they are also partially competitive agents. We believe that this competitive goal is not only acceptable, but also useful for the society, because it induces their agents to improve themselves and hence to improve the performance of the whole society. We define a utility function to represent these individual and social preferences, but some previous definitions are needed.

The return of a coach i in a time t is defined as $R(\omega_i, t) = \frac{V(\omega_i,t)-V(\omega_i,t-1)}{V(\omega_i,t-1)}$. The expression ω_i refers to resources (money and assets) allocated to agent i and function V gives its monetary value, according to the current price. The allocation ω_i may be defined as a tuple $<m_i, \omega_i^1, \omega_i^2, ..., \omega_i^n>$, where $m_i \in \Re$ defines the amount of money allocated to agent i and expression ω_i^j represents the integer number of shares of asset j held by agent i. The monetary value of whole society $V(\omega, t)$ is given by the sum of $V(\omega_i, t)$ of all the coaches $i \in C$ in that society.

Each coach has an expectation about its performance in the near future and this expectation is restricted to the interval $[O_b, O_a]$. We normalize this expectation value to [-1,1], i.e. $-1 \leq Exp_i \leq 1$. The normalized expectation (Exp_i) is used to calculate the expected monetary value for each coach. This expected value is very important to the utility function, because if one agent believes that it will have bad performance it will accept more easily to transfer his resources to other coaches.

We define the expected monetary value $V_e(\omega_i, t)$ for an agent i as $V_e(\omega_i, t) = (1 + Exp_i) * V(\omega_i, t)$. The expected monetary value of the whole society $V_e(\omega, t)$ is given by the sum of $V_e(\omega_i, t)$ of all the coaches $i \in C$ in that society.

As coaches have individual and social preferences, they need to compose both portions to form their utility functions. The relative weight between these portions is modeled as parameter α, where $\alpha \in (0, 1)$ (zero and one are excluded), called *individuality factor*. The value (1-α) is called *social factor*. Whenever $\alpha = 1$, the agent cares only about its own goals, and would be completely individualist. Moreover, the bigger α the less concerned about social preferences would be the agent. In COAST, all coaches are concerned with both criteria, therefore $0 < \alpha < 1$.

We define the utility function $Util_i(\dot{\omega}, t)$ of a coach i as a sum of individual and social preferences weighted by its individual factor. As the negotiation process deals with resource allocation among coaches, utility functions have as parameters the proposed allocation ($\dot{\omega}$), the current allocation (implicit parameter) and a defined instant of time t, as $Util_i(\dot{\omega}, t) = \alpha * UI(\dot{\omega_i}, t) + (1 - \alpha) * US(\dot{\omega}, t)$.

The term $UI(\dot{\omega_i}, t)$ refers to the individual portion of coach preferences. It may be defined as the difference about the expected value of the new allocation and the value of the current allocation, $UI(\dot{\omega_i}, t) = V_e(\dot{\omega_i}, t) - V(\omega_i, t)$.

On the other hand, the social goal is to maximize the return of the whole society portfolio. Therefore, we define one different function to represent the social portion of coach preferences. As the negotiation deals with resource allocation among coaches, the preference function informs if a new allocation $\dot{\omega}$ is preferable over the current allocation ω, according to the social goal. This function is defined as $US(\dot{\omega}, t) = V_e(\dot{\omega}, t) - V(\omega, t)$. Hence, it gives higher numbers for allocations that contribute more to the social goal in a defined instant of time t.

By design, whenever a coach i receives a negotiation proposal for a new allocation, it will accept the proposal if his expected utility is greater or equal to zero, i.e., if $Util_i(\dot{\omega}, t) \geq 0$. The negotiation protocol is described next.

3.3.2 Negotiation Protocol

The negotiation process presented in the coach life cycle (activity 4 on the right of figure 2) is composed by seven sub-activities, which are shown in detail in figure 4. Several coaches interact with each other along the negotiation process, but there is one who is considered the coach with the best performance, who will be named *best coach*. Excluding the best coach, the *bad coaches* are all coaches with negative expectation and the *neutral coaches* are those with positive expectation.

The negotiation process sub-activities are the following :

4.1. **Sends information for others coaches**: Each coach sends to his acquaintances information about its own performance (risk, return and patrimony) and expectation about the near future.

4.2. **Receives information from other coaches**: Each coach receives information from all the others. Hence, each one may calculate the society patrimony, risk and return.

4.3. **Defines coach roles according to performance**: In this activity, each coach calculates the possible roles that each coach, including himself, can play (best, bad or neutral). The coach roles definition is performed by each coach separately, because they are completely autonomous and do not have precedence over the others. However, since we consider that coaches do not lie to each other, they all achieve the same result, since they use the same information. The best coach executes activities 4.6 and 4.7; the others execute activities 4.4 and 4.5.

4.4. ***Other coaches*** - **Receives and analyzes proposals**. This analysis is performed according to individual and social preferences and the current observed situation. Each agent decides if he should accept the proposal or not, based on his utility function as explained in section 3.3.1.

4.5. ***Other coaches*** - **Sends proposals answers**: The proposal answer is sent back to the proponent. If the answer is affirmative, the new allocation is adopted.

4.6. ***Best coach*** - **Prepares and sends proposals**: The best coach prepares a proposal that asks for all available money from the bad coaches.

4.7. ***Best coach*** - **Process proposals answers**: The best coach waits for all the proposal answers. Each affirmative answer creates a deal and the transfer is performed immediately. In case of a negative answer, nothing is changed and the proponent agent does not receive any money from the agent that refused the proposal.

4 Experiments and Results

We have implemented a version of COAST architecture that uses four advisors strategies based on technical analysis. The technical indexes used are the following: moving average, moving average converge-divergence, relative strength index and price oscillator, mentioned in section 2. We have selected 15 assets, which are part of the main index in Brazilian stock market, IBOVESPA, and presented a big number of trading in the last five years (from January, 2006 until December, 2010). Simulation experiments have been performed using a market simulator called AgEx (Castro

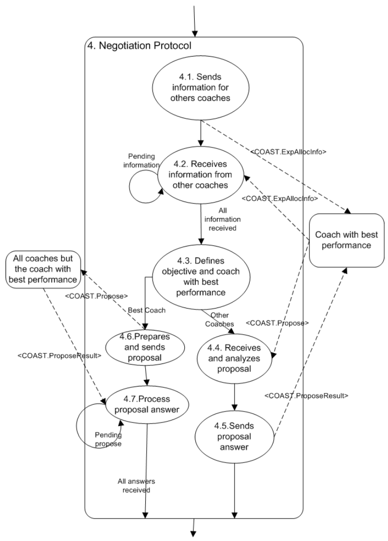

Figure 4. Negotiation process among coaches in COAST architecture.

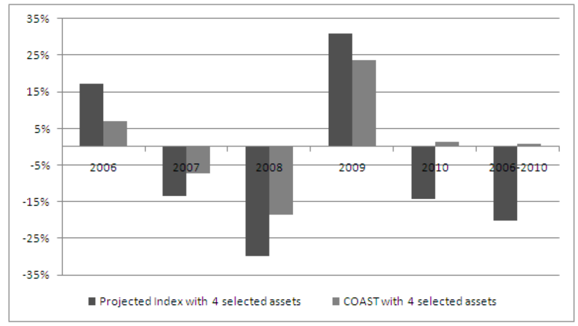

Figure 5. Return achieved by COAST and ***Projected Index*** **using four selected assets**

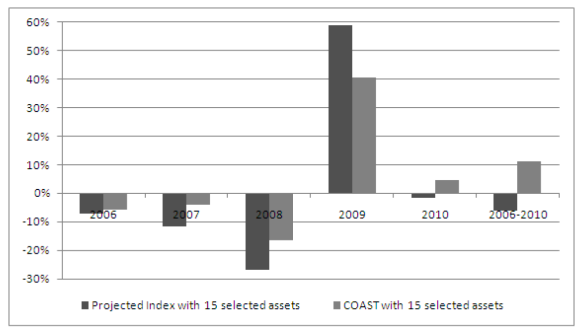

Figure 6. Return achieved by COAST and ***Projected Index*** **using fifteen selected assets**

and Sichman, 2009). We tested COAST societies trading in exchange using daily quotes, where each coach could give one order a day. Additionally, we despised the effect of the orders given by the coaches in the market price, because the agents deal with a very small amount of money when compared to the traded volume for each asset. Despite the fact that the market simulator allows the use of transaction fees, for simplicity we set these fees to zero. In fact, transaction fees have small influence on performance, since there is no big difference in the number of orders given by the analyzed societies (Castro and Sichman, 2009).

In order to analyze COAST performance, we have executed simulation experiments using two different COAST societies, the first with four assets and the second one with all fifteen selected assets. Our first idea of direct comparison was to use the IBOVESPA index. However, a comparison among COAST performance and IBOVESPA would be biased because they do not deal with the same assets. In fact, IBOVESPA index composition changes in time and many assets have been included or excluded along the five years of the evaluation period, i.e. from 2006 to 2010. Due to these facts, we have created a theoretical portfolio called *Projected Index*, which is composed by the fifteen assets used by COAST societies, according to the relative weight of each asset (p_i). Moreover, we normalized these weights to use only the chosen assets using $p_i = \frac{w_i}{\sum_{j \in PI} w_j} * 100\%$, where p_i is the asset weight in Projected Index and w_i is the original weight in IBOVESPA. We have used these weights to define an AgEx trader agent (Castro and Sichman, 2009), which buys and holds a set of shares according to the specified weights. This agent, called also *Projected Index*, acted in the same simulated evaluation period of five years. The figure 5 shows the performance of a COAST society managing four assets against Projected Index with the same four assets. In figure 6, we present the performance of COAST and Projected Index when managing all fifteen assets and finally the performance comparison among the two COAST societies are presented in figure 7.

The comparison among the COAST societies and the *Projected Index* agents in return (figure 5 and 6) shows a better performance of COAST in most of the years and in the whole period (2006-2010). In figure 7, we compare the performance of the COAST societies, the first dealing with four assets and the second with fifteen assets. It is easy to notice that the society with bigger number of assets presented a better performance in all years, except for 2006, and the best performance in the whole period. These facts make us believe that it may be possible to pursuit better results with more assets and that is possible to achieve good performance in using a competitive agent approach.

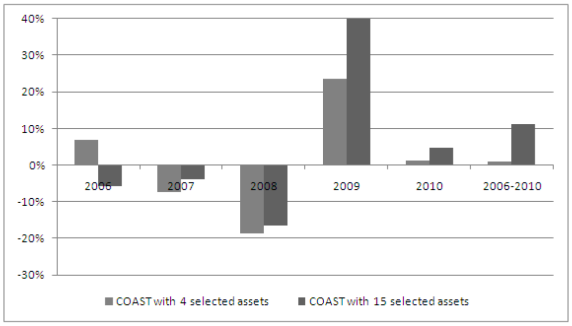

Figure 7. Return achieved by two COAST societies with four and fifteen assets

5 Conclusions and Further Work

In this paper, we presented the COAST multiagent architecture, their agents and a new negotiation mechanism specially designed for this architecture. COAST architecture was implemented and tested in several simulation experiments, which results were presented and analyzed. In the simulated experiments, COAST architecture showed good results and overcome in some scenarios the chosen benchmark (*Projected Index*). It was also possible to realize that is possible to achieve better results using more assets in society.

The main contributions of this work are the *exploitation of competitive strategies* within the COAST architecture (section 3), the fusion of competitive strategies through fuzzy logic and the proposed negotiation mechanism (section 3.3). It also facilitates the use of well known trading strategies as advisors agent's strategies.

In future work, we intend to test COAST architecture with more trading strategies and using a wider evaluation period and number of assets. We believe that a significantly evolution would be a formal modeling of expectations, which are very important in economic reasoning.

Acknowledgments

Jaime Sichman is partially funded by CNPq and FAPESP, Brazil. Paulo Castro thanks Prof. Alexander Brodsky from George Mason University for the basic idea of the *Projected Index* agent.

References

Aldridge, I. (2009). *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*. New York: Wiley.

Araújo, C. H. D. and P. A. Castro (2010). Towards automated trading based on fundamentalist and technical data. In *Proceedings of 20 th SBIA. LNAI*, São Bernado do Campo, Brazil, pp. 704–715. Springer-Verlag.

Castro, P. A. and J. S. Sichman (2007). Towards cooperation among competitive trader agents. In *Proceedings of 9th ICEIS.*, Funchal, Portugal, pp. 138–143.

Castro, P. A. and J. S. Sichman (2009). Agex: A financial market simulation tool for software agents. In W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, J. Filipe, and J. Cordeiro (Eds.), *LNBIP*, Volume 24, pp. 704–715. Berlin: Springer.

Decker, K., A. Pannu, K. Sycara, and M. Williamson (1997, 5–8,). Designing behaviors for information agents. In W. L. Johnson and B. Hayes-Roth (Eds.), *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, New York, pp. 404–412. ACM Press.

Durbin, M. (2010). *All About High-Frequency Trading*. New York: McGraw-Hill.

Feng, X. and C.-H. Jo (2003, 3). Agent-based stock trading. In *proceedings of the ISCA CATA-2003*, Honolulu, USA.

Kearns, M. and L. Ortiz (2003, 11-12). The penn-lehman automated trading project. *IEEE Intelligent System 18*(6), 22–31.

Kendall, G. and Y. Su (2003, 6). Co-evolution of successful trading strategies in a simulated stock market. In *proceedings of The 2003 International Conference on Machine Learning and Applications (ICMLA'03)*, Los Angeles, pp. 200–206.

Pedrycz, W. and F. Gomide (1998). *An introduction to fuzzy sets: analysis and design*. Cambridge, Massachusetts: The MIT Press.

Rosenschein, J. S. and G. Zlotkin (1994). *Rules of encounter:designing conventions for automated negotiation among computers*. Boston: MIT Press.

Sherstov, A. and P. Stone (2004). Three automated stock-trading agents: A comparative study. In *Proceedings of the Agent Mediated Electronic Commerce (AMEC) Workshop - AAMAS 2004*, New York.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. London, England: Wiley.

Yuan Luo, K. L. and D. N. Davis (2002). A multi-agent decision support system for stock trading. *IEEE Network 16*(1), 20–27.