Appl Intell (2013) 38:210–225
DOI 10.1007/s10489-012-0366-8

# Automated asset management based on partially cooperative agents for a world of risks

**Paulo André Lima de Castro · Jaime Simão Sichman**

Published online: 20 July 2012


**Abstract** Despite the fact any investor prefers lower risk and higher return, investors may have different preferences about what would be an acceptable risk or a minimal return. For instance, some investors prefer to have a lower bound risk rather than gaining a higher return. In portfolio theory, it is commonly assumed the existence of one risk free asset that offers a positive return. This theoretical risk free asset combined with a risky portfolio creates a new portfolio that presents a linear relation between risk and return as the risk free asset weight ($w_f$) changes. Hence, any level of risk or of return is easy to achieve separately, just by changing $w_f$. However, in a world without any risk free assets, the combination between assets creates nonlinear portfolios. Achieving a specific level of risk or return is not a trivial task. In this paper, we assume a *risky world* rather than the existence of a risk free asset, in order to model an automated asset management system. Furthermore, some automated asset managers give very different results when evolving in *different contexts*: hence, a very profitable manager can have very bad results in other market situations. This paper presents a multiagent architecture, aiming to tackle these problems. The architecture, named COAST (**CO**mpetitive **A**gent **S**ocie**T**y), is based on *competitive agents* that act autonomously on behalf of an investor in financial asset management. It allows the simultaneous and competitive use of several asset analysis techniques currently applied in the finance field. Some dedicated agents, called *advisors*, apply a particular technique to a single asset. The results provided by these advisors are then submitted to and analyzed by a special agent called *coach*, who evaluates its advisors' performance and defines an expectation about the future price of one specific asset. Within COAST, several coaches *negotiate* to define the best money allocation among different assets, by using a negotiation protocol defined in this paper. We also propose an *investor description model* that is able to represent different investors' preferences with defined acceptable limits of risk and/or return. The COAST architecture was designed to operate adequately with any possible investor's preference. It was implemented using a financial market simulator called AgEx and tested using real data from the Nasdaq stock exchange. The test results show that the architecture performed well when compared to an adjusted market index.

**Keywords** Multiagent systems · Multiagent architectures · Automated asset management · Automated trading

## 1 Introduction

Real world environments may present many different features that may require distinct characteristics for agents evolving in them. There are several different ways to classify environments, one of which proposes that the most complex environments should be those with the following properties [22]: (i) (partially observed) agents do not have complete knowledge about the whole environment state, despite the fact they can estimate it; (ii) (competitive multiagent) some agents compete among themselves and they may change their behavior according to the observation of other agents' actions and the environment state; (iii) (no dominant strategy [25] (p. 77)) there is no known a priori strategy that

P.A.L. de Castro (✉)
Technological Institute of Aeronautics-ITA,
São José dos Campos, SP, Brazil
e-mail: pauloac@ita.br

J.S. Sichman
Intelligent Techniques Laboratory, University of São Paulo,
São Paulo, SP, Brazil
e-mail: jaime.sichman@poli.usp.br

makes an agent overcome the others in all possible scenarios and (iv) (open) at any time, an agent may enter or leave the environment either based on its own decision or because it does not satisfy some minimum requirements to stay in the environment. We call this environment class **complex multiagent environments**. One instance of such class is asset management. Stock market behavior is very hard to predict and agents compete among each other to achieve better financial results. Other possible instances would be strategic games, such as business games or war games, where the choice of the best course of action is dependent on actions and decisions taken by the other players.

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 fact, the stock market may be considered as a special kind of auction, called *continuous double auction*. One the most common kind of orders establishes a target asset, a number of shares, a price for the asset and order type (buy or sell); however, it is also possible to have orders without a pre-defined price, thus relying on the market to define it [5].

This paper presents a multiagent architecture, named COAST (**CO**mpetitive **A**gent **S**ocie**T**y), that is based on *competitive agents* that act autonomously on behalf of an investor in financial asset management. It allows the simultaneous and competitive use of several asset analysis techniques currently applied in the finance field. Some dedicated agents, called *advisors*, apply a particular technique to a single asset; the results provided by these advisors are then submitted to and analyzed by a special agent called *coach*, who evaluates its advisors' performance and defines an expectation about the future price of one specific asset. Within COAST, several coaches *negotiate* to define the best money allocation among different assets, by using a negotiation protocol defined in this paper. We also propose an *investor description model* that is able to represent different investors' preferences with defined acceptable limits of risk and/or return. The COAST architecture was designed to operate adequately with any possible investor's preference. It was implemented using a financial market simulator called AgEx [5] and tested using real data from the Nasdaq stock exchange. The test results show that the architecture performed well when compared to an adjusted market index.

The rest of the paper is organized as follows. In the next section, we briefly present some main concepts related to stock markets and automated asset management. The main contribution of the work, the COAST architecture, is described in Sect. 3. The experiments that we have performed with the architecture are described and analyzed in Sect. 4.

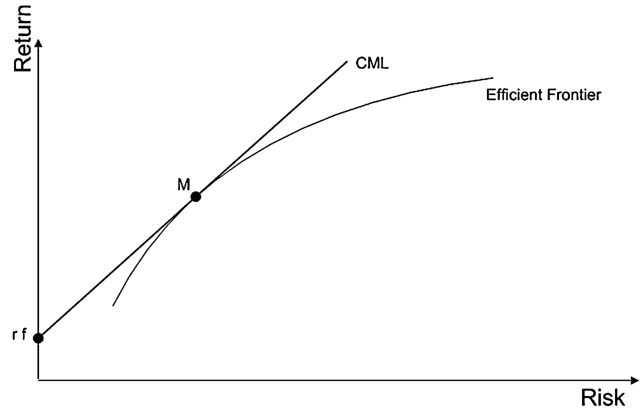



**Fig. 1** CML and efficient frontier in risk versus return graph

Finally, we present our conclusions and further work in Sect. 5.

## 2 Stock markets and automated asset management

In this section, we briefly present some stock markets and automated asset management concepts that were used as a basis for this work, as risk and return measures (Sect. 2.1), technical and fundamentalist analyses (Sect. 2.2) and some common assumptions in automated asset management (Sect. 2.3). We conclude this section by presenting related work in the automated asset management domain (Sect. 2.4).

### 2.1 Risk, return, and efficient frontier

Markowitz [18] proposed a widely known financial theory that models **return** ($R_p$) as a random variable and defines the **risk** as the standard deviation of the return ($\sigma_p$). Therefore, any portfolio with higher variations in its return time series has higher risk than those portfolios with lower variance. In Fig. 1, we show risk (x axis) versus expected return (y axis), where assets are represented by points. The portfolios will always dominate individual assets (i.e., they present a lower risk for the same level of return), due to risk reduction caused by diversification ([12] p. 577). The curve presented in Fig. 1 is formed by the set of portfolios with the lowest possible risk for a given level of return. It is called the **efficient frontier** and it can be shown that it is a nonlinear function [12]. The set of all possible portfolios mixing the available individual assets can be represented by the region in the graph limited by the *efficient frontier*.

Markowitz [18] assumes the existence of a risk free asset ($r_f$) that pays a fixed positive return with the standard deviation of returns equal to zero. This asset $r_f$ is represented by the point $r_f$ in x axis of Fig. 1. If we create a new portfolio $p$ combining the risk free asset $r_f$ and any other portfolio with risky assets $x$, the risk and expected return of portfolio p can be expressed by Eqs. (1) and (2), respectively where $\omega_{r_f}$ is the weight of $r_f$ in the portfolio p and $\sigma_{r_f}$ is its standard deviation [7].

$$E(R_p) = \omega_{r_f} E(r_f) + \omega_x E(R_x) \tag{1}$$

$$\sigma_p = \sqrt{\omega_{r_f}{}^2 \sigma_{r_f}{}^2 + \omega_x{}^2 \sigma_x{}^2 + 2\omega_{r_f} \sigma_{r_f} \omega_x \sigma_x \rho_{r_f x}} \tag{2}$$

However, $r_f$ has a fixed income, therefore the standard deviation of return is zero, and so Eq. (2) can be simplified to Eq. (3). Whether we combine the risk free asset with the portfolio M, we have a set of portfolios represented by the line, called **Capital Market Line** (CML) shown in Fig. 1. The portfolios on CML dominate all others. The portfolios along CML between $r_f$ and M are utilizing the opportunity to invest in the risk free asset, while the portfolios above M are all leveraged; that is, they are borrowing money at the risk free rate. The only portfolio on CML that can be achieved using only risky assets is the portfolio M; therefore it is also on the efficient frontier.

$$\sigma_p = \omega_x \sigma_x \tag{3}$$

The index proposed by Sharpe [23] is commonly used to evaluate risk and return of portfolios. It is presented in Eq. (4) and it is known as the *Sharpe index*. The $r_f$ expression in Eq. (4) refers to return on an asset assumed to be free of risk. In fact, the portfolio M is the portfolio with the highest possible Sharpe Index and CML could be defined as the line that passes through the point $r_f$ with the highest possible slope and at least one point in the efficient frontier.

$$Sharpe(p) = \frac{R_p - r_f}{\sigma_p} \tag{4}$$

Despite some critics [27], the Sharpe index has been widely adopted in finance industry to point out preferable portfolios and even to compare fund managers performances. Frequently, investors are implicitly modeled in the finance field as rational agents that try to maximize their Sharpe index.

In order to achieve its goal, a manager adopts a certain *trading strategy*, which may be defined as a function $\mu : S \to \theta$ that associates the current *market state* $s \in S$ to the order $\theta_i \in \theta$ that should be submitted to the stock market. A market state contains all available information about the market at a given moment.

### 2.2 Technical and fundamentalist analyses

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. Examples of such strategies are *moving average* and *moving average converge-divergence*, both based on the average of the last price of the target asset; *stochastic* and *relative strength index (RSI)*, based on price variation; *price oscillator* and *price volume trend*, based on the identification of price trend and negotiation volume [4].

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 growth rates, and global growth rate. 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 many fundamentalist concepts in an algorithm, although it is much easier to design algorithms to calculate time series used in technical analysis [2].

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 the market scenario [4]. This observation determined the first guideline for our architecture, i.e., to facilitate the composition of different strategies, as shown in Sect. 3.2.

### 2.3 Common assumptions in automated asset management

In the literature related to automated asset management, authors usually use data collected from real stock markets to perform simulated experiments. This choice is based on the implicit assumption that if an agent may achieve good performance in such experiments, he will likely achieve good performance in actual trading. This assumption makes sense, but new designers should be careful with some possible problems that may lead to mistakes about real agent's performance. These common assumptions are the following:

- **Influence of the new agent in the market**: It is usually assumed that as one agent has a very small amount of shares compared to total volume traded in the market, the effect of his orders on price formation will be very small. Hence, authors very often despise such influence. However, there are professional investors that try to copy the trading strategies of the most successful managers. Therefore, whenever an agent achieves good results, the orders based on his strategy will probably affect the market by the shares managed both by him and by his followers. When the number of followers increases, despising the direct influence of the leader agent may become a flawed assumption.
- **Relevance of database**: A database for experimental simulations is defined by the set of chosen assets and the period of time considered for the data gathered from real markets. It is important to observe that creating a trading strategy for a previously known database is trivial

and consists of buying each asset at a low price and selling it at a high price. The great challenge in automated asset management is to build an automated system able to operate without previously knowing the database. Designing an automated system using a very small or fixed database for assessment of the system's performance may lead to risk of over-fitting, i.e., the system has good performance only in markets whose situation is similar to the used database. In fact, if an automated trading system is excessively adapted to specific situations, it may present very low performance when presented to new situations, as pointed out in [13]. It is not rare that a trading strategy performs very well in a certain database and very badly in another one [24]. On the other hand, it is quite hard to define what a relevant database is. Since it is not trivial to define a sufficient set of assets and a period of time long enough to capture long term tendencies, designers should be careful and use several different time periods for adjusting and testing their systems, including good and crisis periods, as well as experimenting with different sets of assets.

- **Risk free asset assumption**: The existence of a risk free asset that pays a fixed positive return is a very common assumption in the finance field [18]. Despite this fact, no asset presents standard deviation of returns equal to zero in reality. Usually, it is assumed that US Treasury bills are risk free. In this paper, we assume a *world of risks* and therefore we do not assume the existence of any risk free asset. We assume that there are very low risk assets, but they are not risk free. Therefore, the relation between expected return and risk of any portfolio with two assets is a non-linear function with no *capital market line* (CML). That assumption would make a mathematical approach cumbersome, but in an agent-based approach such a problem does not happen, because agent-based simulations can handle a wide range of nonlinear behavior, as stated by [10]. Even without the existence of a risk free asset assumption, we have the efficient frontier as a set of possible portfolios, but it is not possible to choose a portfolio on efficient frontier just by combining the M portfolio and lending or borrowing money at the risk free asset rate, as on the CML.

### 2.4 Related work

Artificial Intelligence is progressively gaining relevance in the financial world [19]. In fact, Automated asset management, also known as automated stock trading, algorithm trading, or high frequency trading, has been a focus for many researchers. It is possible to classify these initiatives in many different ways [5]. One particularly interesting classification concerns the *number of assets* dealt with simultaneously: many assets (multi-asset) or just a single asset (mono-asset). It should be pointed out that the management of several assets is more complex than just the creation of several instances of one asset manager. It is necessary to explore the complementarities among the group of assets, especially to minimize the portfolio risk. As pointed out in [5], there is more in the literature in the second group (mono-asset) than in the first group. Another classification concerns the *trading strategy criteria*. One can easily verify that there is more work concerned exclusively with return criterion [3, 6, 11, 14, 15, 24, 28, 31] than with risk criterion. In fact, risk criterion is typically approached as a trade-off solution, combined with return in the Shape index [4, 16], and not as an independent dimension, as addressed in this work. Another relevant distinction deals with the typical *time interval between orders*. Some researchers and practitioners try to achieve better performance exploring the fact that an automated system can analyze a significantly higher amount of information when compared to a human being, and hence the time interval can be reduced from days or hours to a few seconds or even some fractions of a second. This kind of work is often called high frequency trading, and tries to achieve better returns with very fast changes in the current position [1, 8, 17].

Except for [4], we may notice that none of those papers explores *competition* among trading strategies, as we propose in this work. Another characteristic of most current automated asset management systems is the fact that they do not adapt their behavior to possible different investors' *preference patterns*. For instance, for some investors the property of having a lower bound risk may be even more important than gaining a higher return. Additionally, one can notice that some automated asset managers show very different results when evolving in *different contexts*. A very profitable manager can show very bad results in other market situation. The COAST architecture, presented next, aims to tackle these problems.

## 3 Coast architecture

In this section, we present the main features of the COAST architecture: the investor description model (Sect. 3.1), the agents (Sect. 3.2) and the negotiation mechanism specially developed for the architecture (Sect. 3.3). At the end of the section, we also present a theoretical evaluation of this mechanism (Sect. 3.4).

### 3.1 Investor description model

In a world of risk as we assume here, it is not enough to determine one point in the efficient frontier and make linear combinations with $r_f$ to satisfy investors with different levels of acceptable risk and return, because the absence of risk

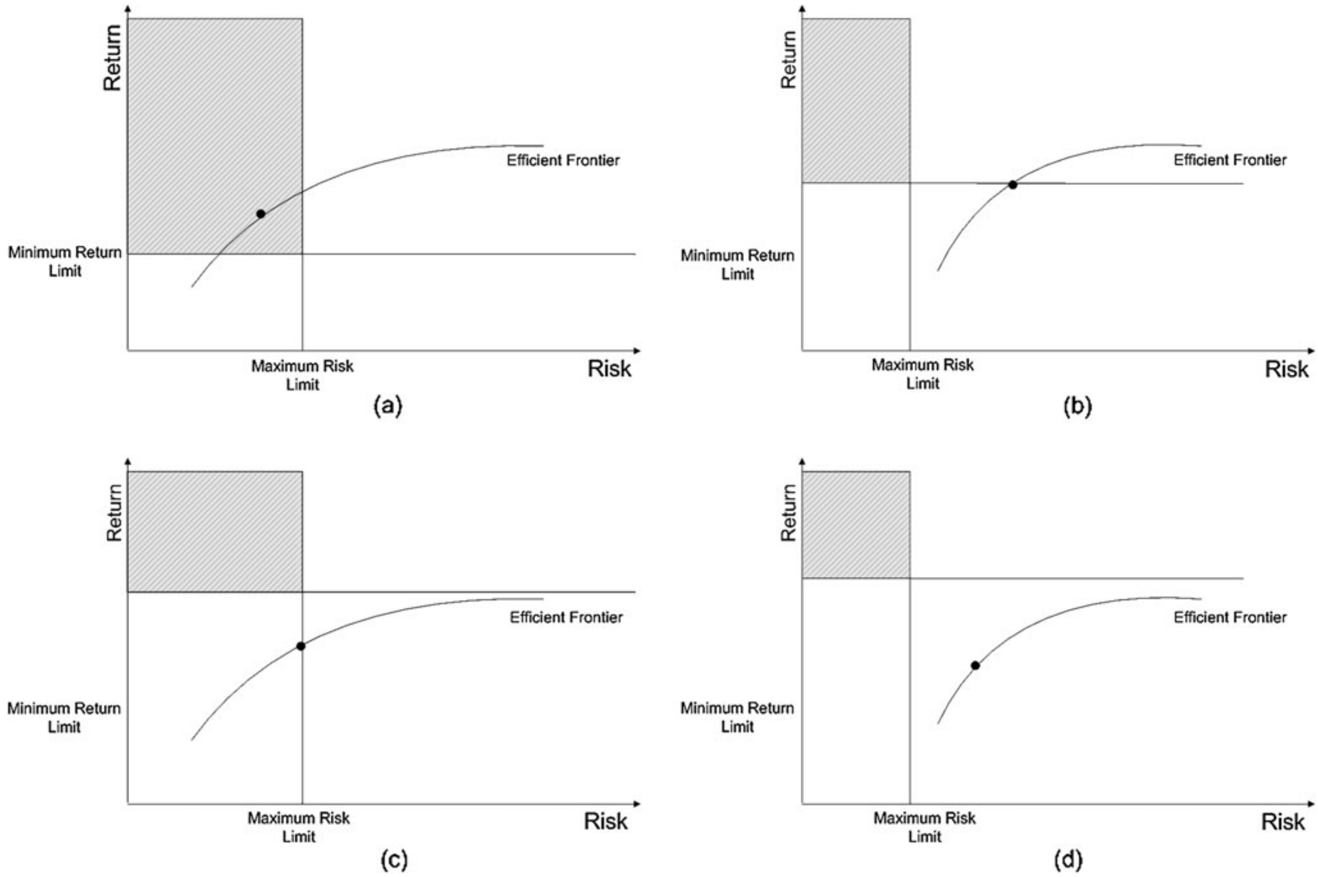



**Fig. 2** Possible market and investor preferences situations represented in a risk versus return graph. Investor preferred region is represented by the *gray area*

free assets eliminates the CML. A manager would need to determine the non-linear function that describes the efficient frontier in order to achieve specific levels of risk and return, as discussed in Sect. 2.1. In fact, investors may have different levels of acceptance for return and risk criteria and automated manager needs to be aware of investors' preferences. In order to model these preferences, we create an investor description with two explicit parameters: *maximum acceptable risk* and *minimum acceptable return*. As all investors are averse to risk, there is no sense in creating a minimum limit of risk. Furthermore, as all investors desire higher return, there is no sense in creating a maximum limit of return.

It is possible to represent such preferences through acceptable regions in a risk versus return graph as shown in Fig. 2. Figure 2 also shows the *efficient frontier* [18]. The efficient frontier is the set of portfolios with the highest return to a given level of risk [12]; the portfolio represented by the indicated point in Fig. 2 is the one that presents the highest Sharpe Index of all portfolios in efficient frontier and that is also in the indicated region. As COAST deals only with risky assets, the investor's goal in the situation in Fig. 2(a) is to reach the indicated point.

Regardless of the investor parameters, a trading agent will pursue **only one** of three different *social goals* at a given moment: *to minimize risk* (*G1*), *to maximize return* (*G2*) or *to maximize the chosen trade-off solution* (*G3*), in our case using the Sharpe index. The decision is taken according to the current levels of return and risk and to the investor parameters in a quite straightforward way: if the risk is above the acceptable level (Fig. 2(b)), the manager should then act in order to minimize risk. Similarly, the trading agent can choose to maximize return if it is below the acceptable level (Fig. 2(c)). However, when both risk and return are currently unacceptable (Fig. 2(d)), the trading agent should decide to maximize the trade-off solution in order to try to change both return and risk simultaneously. In Table 1, we show the possible social goals adopted by the agent society, according to the current market situation.

The main concern is not really to adopt and keep one particular portfolio, but to keep changing the portfolio in order to adjust it to continuously changing market conditions. Moreover, these choices must be made according to the investor's preferences, represented by his profile, in order to stay in the indicated regions and move towards the goal points. As discussed in Sect. 1, most of the work in automated asset management deals with implicit investor preference, which is always to maximize the expected return. Such situations may be described in our model as an investor with a very high (or infinite) value for minimum acceptable return. A few other studies try to maximize a tradeoff solution between risk and return [4, 16]. Such systems may be described in our model as an investor with low value for minimum acceptable return and a very high value for maximum acceptable risk.

## 3.2 COAST agents

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 for the society's owner, the investor is described through our investor description model. These trading strategies are materialized through agents called *advisors*. In COAST, strategy outputs are not interpreted as orders, but as advice about one specific asset. The other architectural guidelines are the following: (i) to work with many different assets, (ii) to adapt strategy's relevance to each asset and (iii) to avoid central agents or a centralized decision making procedure about resource allocation through assets. In fact, there are multiple coordinator agents, called *coaches*. Each coach is specialized in one specific asset and they are cooperative; they negotiate about which would be a preferred portfolio for the user. Therefore, a society with four assets and three different strategies would be composed of four coaches and twelve advisors (three advisors for each coach), as shown in Fig. 3. The advisors are competitive agents and they communicate only with their respective coach. Therefore, there is one group of cooperative agents and another group of competitive agents (Fig. 3). The advisors located in the same column operate with the same asset and the coach at the top of the column evaluates and coordinates the work of the advisors in that column. According to these evaluations, one advisor with good performance has more relevance in the coach's decisions than the other advisors. Coaches auto-evaluate and negotiate to allocate more money to the coaches with better performance in the society, as described in Sect. 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.

Coaches know each other and are also aware of the risk and return preferences of the investor. On the other hand, the advisors are concerned just with the asset return and try to inform the investors of the right moment to buy or sell the asset. These two kinds of agents are described in detail next.

**Table 1** Social goals according to current market situation

| Market situation | Social goal |
|---|---|
| Acceptable risk and return | Maximize Sharpe index |
| Unacceptable risk and acceptable return | Minimize risk |
| Acceptable risk and unacceptable return | Maximize return |
| Unacceptable risk and return | Maximize Sharpe index |

### *3.2.1 Advisors*

Advisor agent suggests buying or selling a number of shares of a specific asset following their own strategy. Advisors can

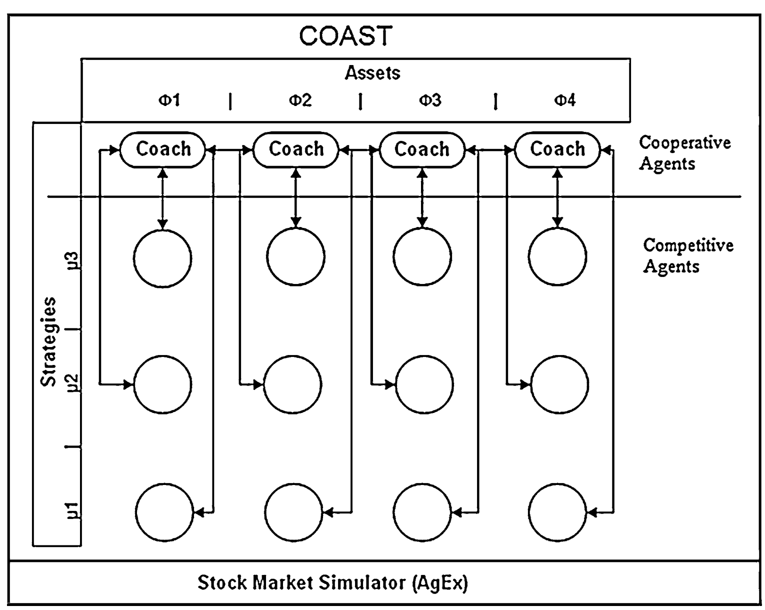



**Fig. 3** Example of a **CO**mpetitive **A**gent **S**ocie**T**y that manages four assets using three strategies simultaneously

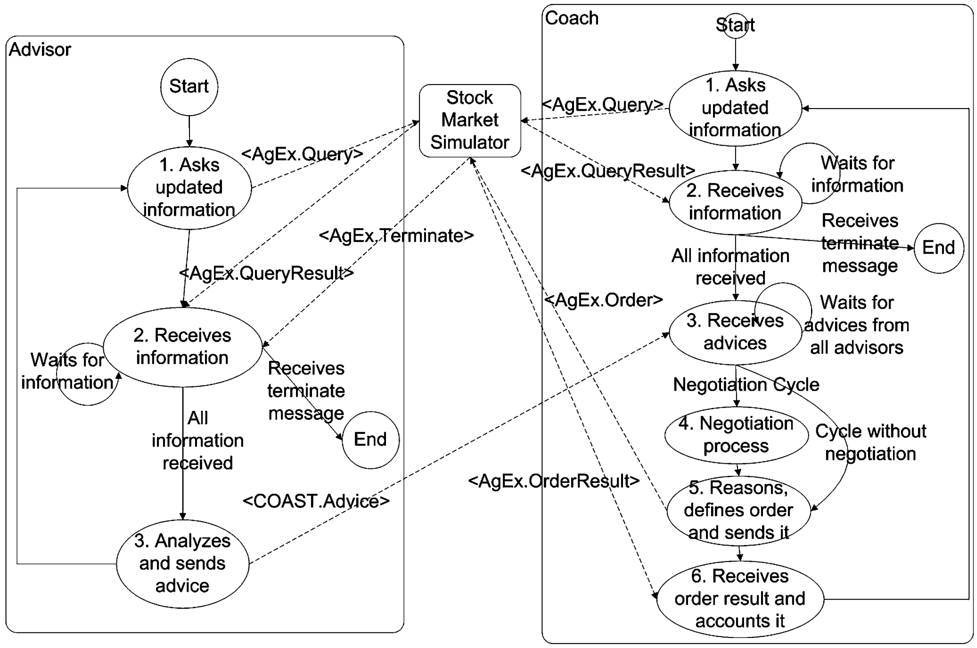



**Fig. 4** Advisor and coach life cycles in COAST architecture

be easily created using any well known trading strategy. This advice is sent to the coach, who is the agent in charge of order definition. An agent life cycle may be described as the set of activities that the agent performs while still active. The advisor's life cycle is presented on the left of Fig. 4. 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* [5], which can be seen in the center of Fig. 4;
2. **Receives information**: The stock market simulator returns the information which is locally stored. This step is also used to synchronize all agents in simulated time [5];
3. **Analyses and sends advice**: According to the collected information and his strategy, the advisor defines and sends buy/sell/hold advice to his coach.

The coach evaluates its advisors' performance according to their advice and the market evolution. For instance, whenever an advisor suggests buying an asset whose price rises after the advice, this advisor is positively evaluated. A similar reasoning can be made regarding selling advice. The advisor's goal is to achieve the highest possible evaluation from his coach, because this situation gives him more relevance to the whole society. The coach keeps a memory of the recent advice and its status, in order to compute his advisors' evaluations.

### 3.2.2 *Coaches*

Coaches *receive* advice, *evaluate* their advisors, *negotiate* with other coaches and *define orders* that are submitted to the market. These activities are performed along the entire agent's 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 the COAST society parameters.

The coach activities are presented on the right of Fig. 4, and have the following meaning:

1. **Asks for updated information**: The coach asks for new information about its target asset. Even though coach orders are completely based on the messages sent by his advisors, he would need asset information to evaluate them;
2. **Receives information**: The *stock market simulator* [5] returns the stored information to be used in the advisors' evaluation;
3. **Receives advice**: The coach receives advice from each advisor that deals with the asset that he manages. This activity continues until he receives advice from all his advisors. During this activity, he also performs the advisors' evaluation, as described in Sect. 3.2.1. If the current cycle must include negotiation, the next step is activity 4; otherwise, the next step is activity 5;
4. **Negotiation Process**: The coach negotiates with other coaches in order to define a new resource (money) allocation. This activity is much more complex than the others and therefore is described in more detail in Sect. 3.3;

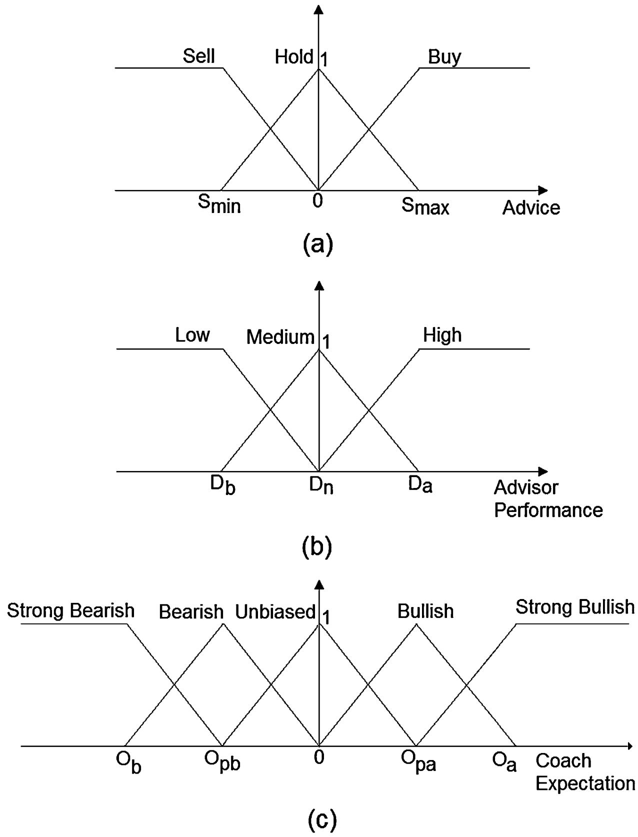



**Fig. 5** Membership functions of fuzzy variables **advice**, advisor **evaluation** and coach **expectation**

5. **Reasons, defines orders, and sends them**: Based on advisors' suggestions and following a set of fuzzy rules which are presented in the sequence, the coach defines his order and sends it for execution to the *stock market simulator* [5];
6. **Receives order result and registers it**: The coach receives the order result, whose value may be total, partial or not executed at all, and registers it in his portfolio, including the real price that was used to buy or sell the asset.

Coaches calculate their expectation about their own performance in the near future. This performance **expectation** is modeled as a linguistic variable [20] with five terms: *strong bearish*, *bearish*, *unbiased*, *bullish* and *strong bullish*, as shown in Fig. 5(c). The **advice** is a linguistic variable, with three linguistic terms: *sell*, *hold*, and *buy*, whose membership functions are presented in Fig. 5(a). Meanwhile, 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 %], as shown in Fig. 5(b). The coach determines his expectation based on the information (advice) that comes from his advisors and their respective evaluation according to set a fuzzy rules, as seen in Fig. 6.

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** *Hold*
**Then** *Expectation* **is** *Unbiased*
R6. **If** *Evaluation* **is** *Low*
**Then** *Expectation* **is** *Unbiased*

**Fig. 6** Coach expectation fuzzy rules

The coach expectation ($Expectation_i$) is both used in the negotiation mechanism with other coaches, described in Sect. 3.3, and also in the definition of his order. The coach defines his order decoding his fuzzy expectation to a crispy value; in our implementation, the fuzzy decoding method used was the center of gravity method [20]. For instance, if the coach expects a *strong bullish* market, it 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 keep his current position.

### 3.3 Negotiation mechanism

Coaches have individual and social preferences and negotiate according to those preferences. A negotiation mechanism is defined by a *negotiation protocol*, composed of the *communication rules* among participants, and by the players' *strategies* [21]. There are some well known negotiation protocol, probably the most known is the contract net protocol [26]. Another possible example is the monotonic concession protocol [21]. However, contract net is defined for task-oriented domains and the one-step protocol is defined for worth-oriented domains [21, 30]. The domain addressed by COAST, asset management, does not deal with task allocation, but financial asset allocation according to agent's expectations and preferences. It cannot be classified as a task-oriented domain. Despite the fact, one can argue it can be classified as worth-oriented domain. However, there is an important issue that makes the definition of a worth function very hard: the state's worth depends on what is going to happen in the future with stock prices, for example. Therefore, we designed a new negotiation protocol and strategy for our agents.

In this section, we describe the individual and social preferences of COAST agents, then the proposed negotiation protocol and finally the roles that a coach can assume in each negotiation round and the strategy adopted for them.

#### 3.3.1 Individual and social preferences

As previously presented in Table 1, each coach has a global social goal to achieve. However, competitive agents try to overcome the others and get more resources for themselves. We believe that this competitive goal is not only acceptable, but also useful for the society, because it induces agents to improve themselves and, hence, to improve the performance of the whole society.

In order to define a utility function to represent these individual and social preferences, some previous definitions are needed. The return of a coach $i$ in a time $t$ is defined as:

$$R(\omega_i, t) = \frac{V(\omega_i, t) - V(\omega_i, t-1)}{V(\omega_i, t-1)} \tag{5}$$

In Eq. (5), $\omega_i$ refers to resources (money and assets) allocated to agent $i$. The allocation $\omega_i$ may be defined as a tuple $\langle m_i, \omega_i^1, \omega_i^2, \ldots, \omega_i^n \rangle$, where $m_i \in \Re$ defines the amount of money allocated to agent i and expression $\omega_i^j$ represents the integer number of shares of asset $j$ held by agent $i$. Furthermore, $V(\omega_i, t)$ is the monetary value of $\omega_i$ at time $t$.

The risk $K(\omega_i, t)$ is defined as the standard deviation of the return $R(\omega_i, t)$, which can be estimated using the $N$ last observed returns through Eq. (6):

$$K(\omega_i, t) = \sqrt{\frac{\sum_{j=1}^{N} (R(\omega_i, t_j) - \bar{R}(\omega_i))^2}{n-1}} \tag{6}$$

The return $R(\omega, t)$ and risk $K(\omega, t)$ associated to the whole society may be defined similarly. They are defined in Eq. (7) and Eq. (8), respectively. The monetary value of the 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:

$$V(\omega, t) = \sum_{i \in C} V(\omega_i, t) \tag{7}$$

$$R(\omega, t) = \frac{V(\omega, t) - V(\omega, t-1)}{V(\omega, t-1)} \tag{8}$$

$$K(\omega, t) = \sqrt{\frac{\sum_{j=1}^{N} (R(\omega, t) - \bar{R}(\omega))^2}{n-1}} \tag{9}$$

Each coach $i$ has an expectation ($Expectation_i$, as described in Sect. 3.2.2) about its performance in the near future and this expectation is restricted to the interval $[O_b, O_a]$. We normalize this expectation interval to $[-1, 1]$, i.e. $-1 \le Exp_i \le 1$, through Eq. (10). The normalized expectation ($Exp_i$) is used to calculate the expected monetary value for each coach. This expected value is a very important input to the utility function, because if one agent believes that he will have bad performance he will more easily accept to give his resources to other coaches.

$$Exp_i = \frac{2 * Expectation_i - O_a - O_b}{O_a - O_b} \tag{10}$$

We define the expected monetary value $V_e(\omega_i, t)$ for an agent $i$ as the current monetary value plus the expected change, as stated in Eq. (11). 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 shown in Eq. (12):

$$V_e(\omega_i, t) = V(\omega_i, t) + Exp_i * V(\omega_i, t) \tag{11}$$

$$V_e(\omega, t) = \sum_{i \in C} V_e(\omega_i, t) \tag{12}$$

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 a parameter $\alpha$, called *individuality factor*, where $\alpha \in [0, 1)$ (zero is included, 1 is excluded). The value $(1-\alpha)$ is called the *social factor*. Whenever $\alpha = 1$, the agent cares only about its own goals, and would be completely individualistic. A bigger $\alpha$ means that an agent is less concerned about social preferences. 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 the sum of individual and social preferences weighted by its individual factor as stated in Eq. (13). 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 $t$:

$$Util_i(\dot{\omega}, t) = \alpha * UI(\dot{\omega_i}, t) + (1-\alpha) * US(\dot{\omega}, t) \tag{13}$$

The term $UI(\dot{\omega_i}, t)$ refers to the individual portion of coach preferences. It may be defined as the difference between the expected value of the new allocation and the value of the current allocation, as presented in Eq. (14):

$$UI(\dot{\omega_i}, t) = V_e(\dot{\omega_i}, t) - V(\omega_i, t) \tag{14}$$

We define three different functions to represent the social portion of the coach's preferences for risk minimization (G1), return maximization (G2) or Sharpe index maximization (G3), one for each possible social goal. As the negotiation deals with resource allocation among coaches, the preference function informs whether a new allocation $\dot{\omega}$ is preferable over the current allocation $\omega$, according to the current social goal. These functions are called $US_{G1}(\dot{\omega}, t)$, $US_{G2}(\dot{\omega}, t)$ and $US_{G3}(\dot{\omega}, t)$ for social goals G1, G2 and G3, respectively. Each one gives higher numbers for allocations that contribute more to the social goal in a defined instant of time $t$ as formally defined in Eqs. (15), (16) and (17), respectively:

$$US_{G1}(\dot{\omega}, t) = K(\omega, t) - Max\_Risk \tag{15}$$

$$US_{G2}(\dot{\omega}, t) = V_e(\dot{\omega}, t) - V(\omega, t) \tag{16}$$

$$US_{G3}(\dot{\omega}, t) = Sharpe_e(\dot{\omega}, t) - Sharpe(\omega, t) \tag{17}$$

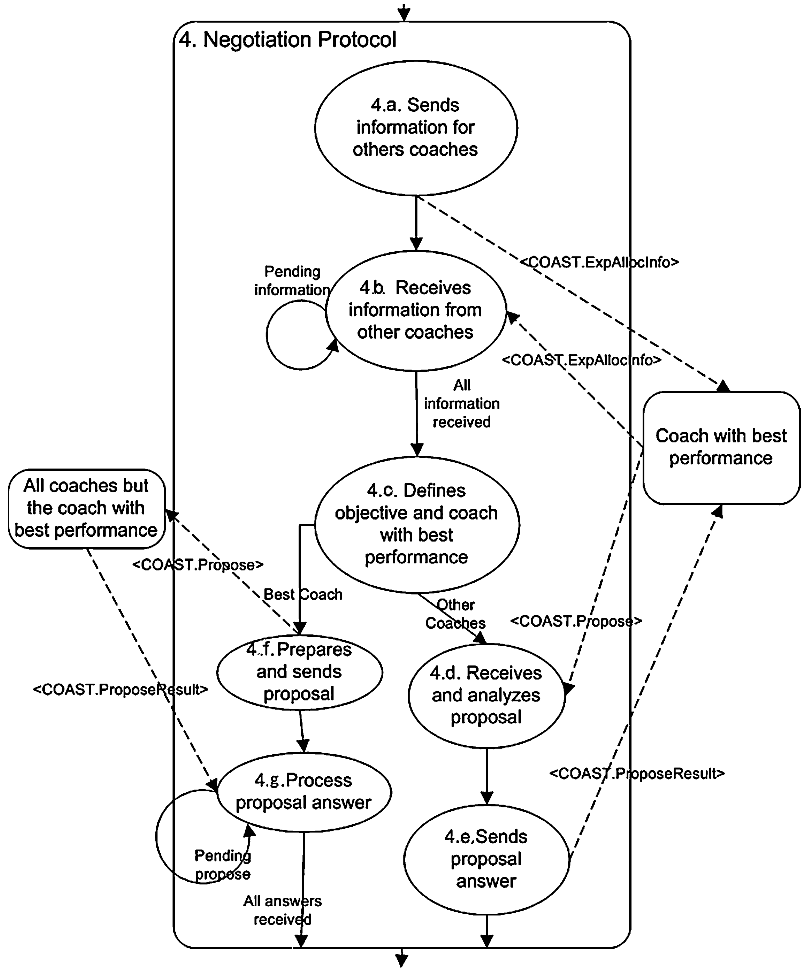



**Fig. 7** Negotiation process among coaches in COAST architecture

In Eq. (15), the expression *Max_Risk* refers to maximum acceptable limit of risk, according to the investor description model (Sect. 3.1). The social preference function $US_{G3}(\dot{\omega}, t)$ is based on the Sharpe index, expressed in Eq. (18). The expected Sharpe index for the proposed allocation is presented in Eq. (19):

$$Sharpe(\omega, t) = \frac{\overline{R}(\omega) - r_f}{K(\omega, t)} \tag{18}$$

$$Sharpe_e(\dot{\omega}, t) = \frac{\overline{R}_e(\dot{\omega}) - r_f}{K(\omega, t)} \tag{19}$$

$$\overline{R_e}(\omega) = \frac{\sum_{i=1}^{N} \overline{R(\omega_i)} * (1 + Exp_i)}{N} \tag{20}$$

The expressions $\overline{R}(\omega)$ and $\overline{R_e}(\omega)$, Eq. (20), are simple **averages** of $R(\omega)$ and $R_e(\omega)$, respectively. The symbol $r_f$ refers to the return of the risk-free asset [23].

In summary, whenever a coach $i$ receives a negotiation proposal for a new allocation, he will accept the proposal if his expected utility, Eq. (13) 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 of the right of Fig. 4) is composed of seven sub-activities, which are shown in detail in Fig. 7. Several coaches interact with each other throughout the negotiation process, but one of them, named *best coach*, is considered the coach with the best performance. From sub-activity 4.c on, the behavior of the best coach and the others differ.

The negotiation process sub-activities are the following:

4.a. **Sends information to others coaches**: Each coach sends to other coaches information about his own performance (risk, return and patrimony) and expectation about the near future;

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

4.c. **Defines coach roles according to performance**: In this activity, each coach calculates the possible roles that each coach, including himself, can play. There are three possible roles: best coach,[1] neutral coach or bad coach, as detailed in Sect. 3.3.3. 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.f and 4.g; the **other coaches** execute activities 4.d and 4.e;

4.d. ***Other coaches*—Receives and analyzes proposals**: This analysis is performed according to individual and social preferences and the current observed situation. The coach decides if he should accept the proposal or not, based on his utility function as explained in Sect. 3.3.1;

4.e. ***Other coaches*—Sends proposal answers**: The proposal answer is sent back to the proponent. If the answer is affirmative, the new allocation is adopted;

4.f. ***Best coach*—Prepares and sends proposals**: The best coach prepares a proposal that asks for all the available money from the underperforming coaches, according to the society goals. For instance, if the goal is to reduce risk, all coaches with current risk higher than the limit defined by the investor receive a proposal. The proposal creation process is described in Sect. 3.3.3;

4.g. ***Best coach*—Process proposal 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 best coach does not receive any money from the agent that refused the proposal.

The definition of the roles that each coach plays, according to the social goal, is detailed next.

#### 3.3.3 Coach roles

As described in Sect. 3.1, the social goal can be one the following possibilities: risk minimization (G1), return maximization (G2) or Sharpe maximization (G3). Given the social goal, coach roles are formally defined as follows:

- Social goal: Risk minimization (G1):
  - *Best coach*: The coach with lowest risk, in the case of a tie one of them is randomly chosen as the best one;
  - *Bad coach*: Any coach $i$ that $K(\omega_i, t) > Max\_Risk$;
- Social goal: Return maximization (G2):
  - *Best coach*: The coach with highest *expectation*, in the case of a tie one of them is randomly chosen as the best one;
  - *Bad coach*: Any coach $i$ that $Exp_i < 0$;
- Social goal: Sharpe maximization (G3):
  - *Best coach*: The coach with highest trade-off (see Eq. (18)); in the case of a tie one of them is randomly chosen as the best one;
  - *Bad coach*: Any coach $i$ that $Exp_i < 0$.

No matter the social goal, *neutral coaches* are all coaches that are neither chosen as the *best coach* nor have the requisites to be considered a *bad coach*. The set $D$ is defined as the set of all bad coaches.

The best coach $v$ creates several proposals and sends one proposal to each one of the bad coaches, i.e. $\forall i \in D$. The proposal brings a new allocation ($\dot{\omega}$) derived from the original allocation ($\omega$), where the best coach asks for all the money from all bad coaches, therefore $\dot{m}_i = 0, \forall i \in D$ and $\dot{\omega}_v = \omega_v + \sum_{j \in D} m_j$.

There are no changes for neutral coaches. The bad coaches may, however, deny the transfer if they prefer to. This happens when the new allocation utility calculated by Eq. (13) is less than zero; however, bad coaches accept proposals many times because the social part of their utility functions overcomes the individual part.

### 3.4 Negotiation mechanism theoretical evaluation

The COAST negotiation mechanism was designed to represent correctly both the agent's individual and social preferences, and also to present some desirable features for negotiation mechanisms, as the ones pointed out by Sandholm [29] (Chap. 5) and by Wooldridge [30]. Despite other features that could be considered [30], we selected the following features to characterize our approach:

- **Guaranteed success**: The COAST mechanism never reaches a deadlock, because in the case of no acceptance of a proposal, the current allocation is kept;
- **Maximizing expected social welfare**: The mechanism determines that any accepted allocation $\dot{\omega}$ has a bigger sum of agent's utilities than a previous allocation $\omega$, as shown in Theorem 1;
- **Simplicity**: The coach strategy is very simple. He asks the best possible allocation if he has the right to propose and accepts a proposal only if it is indifferent (zero utility) or good (positive utility), as described in Sect. 3.3;
- **Computational efficiency**: As stated by Sandholm [29] (p. 204), mechanisms with lower computational cost are preferable to those with higher demand for processing power, except when it can be shown that the additional computational complexity may be justified by a significant higher quality in the solution. The COAST mechanism requires fewer calculations to be performed by each coach;
- **Distribution**: There is no single point of failure or any special agent that coordinates the others. The conflicts of interest are solved by negotiation among coaches.

[1] There is only one single best coach per negotiation cycle; in the case of tie, one of them is randomly selected.

These features were chosen because they seem to be more adequate to our context, due to their emphasis on social preferences. They also allow some deals where some agents may lose all their resources in particular situations. That kind of deal may be better to the society, even if it is very bad for those agents that loose all their resources. Such deals would not be possible if the system tried to maximize the utility achieved by the worst coach, instead of maximizing the social welfare. Therefore, we believe that maximizing the social welfare is a better solution in our context.

At each negotiation round, the COAST negotiation mechanism produces one allocation deal that either has a higher social welfare (if a coach accepts a proposal) or social welfare remains the same (if no proposal is accepted), as shown by Theorem 1.

**Theorem 1** *Any new accepted allocation $\dot{\omega}$ presents higher expected social welfare than current allocation $\omega$.*

*Demonstration* Let $\dot{\omega}$ be a new allocation accepted by a COAST society and $\omega$ the current allocation. Then for any agent $i \in D$, $Util_i(\dot{\omega}, t) \geq 0$, where $D$ is the set of bad coaches, because no agent accepts a proposal with utility lower than zero. Given the utility function definition in Eq. (13), it is easy to see that the utility of the current allocation $\omega$ is zero. Therefore $\forall \in C, Util_i(\dot{\omega}, t) \geq Util_i(\omega, t) = 0$, where $C$ is the set of all coaches. The best coach (v) always receives more money and then its utility is higher than the previous utility $Util_v(\dot{\omega}, t) > Util_v(\omega, t)$ such that we can deduce that $\sum_{i \in C} Util_i(\dot{\omega}, t) > \sum_{i \in C} Util_i(\omega, t)$.

It is important to notice that Theorem 1 does not guarantee that the allocation proposed in the deal has the highest utility among all possible allocations. In order to guarantee that, it would be necessary to search for all possible allocations, which would have a very high computational cost, since the possible allocations number grows exponentially with the amount of the society's money. Moreover, this search would be senseless, because the social welfare may change over time and spending a lot of time to determine the best result could be quickly outdated. Furthermore, it is also important to note that Theorem 1 does not guarantee that society do not face the risk of losing monetary value, but that the society will only accept deals that present higher expected social welfare.

## 4 Experiments

We have implemented a version of the COAST architecture that uses six advisor strategies based on technical analysis. The technical indexes used are the following: moving average, moving average converge-divergence, stochastic, relative strength index, price oscillator and price volume trend, which were briefly described in Sect. 1. COAST has been implemented over JADE platform [9] and Java language.

**Table 2** List of assets identificators from Nasdaq exchange used in the simulated experiments

| | Asset ID | | Asset ID |
|---|---|---|---|
| 1 | AAPL | 18 | FISV |
| 2 | ADBE | 19 | INTC |
| 3 | ALTR | 20 | KLAC |
| 4 | AMAT | 21 | LLTC |
| 5 | AMGN | 22 | LRCX |
| 6 | APOL | 23 | MSFT |
| 7 | CDNS | 24 | MXIM |
| 8 | CELG | 25 | ORCL |
| 9 | CMCSA | 26 | PAYX |
| 10 | COST | 27 | PCAR |
| 11 | CSCO | 28 | ROST |
| 12 | CTAS | 29 | SPLS |
| 13 | DELL | 30 | SYMC |
| 14 | ERIC | 31 | TEVA |
| 15 | ERTS | 32 | TLAB |
| 16 | EXPD | 33 | XLNX |
| 17 | FAST | 34 | XRAY |

Since we developed a system that reallocates resources among agents that manage a portfolio of financial assets, some questions may naturally arise. For example, is it a good idea to reallocate resources among coaches during the trading period? In the case of a positive answer, should the system reallocate resources among coaches often or infrequently? Does the system really present a good performance?

In order to analyze these questions, we designed a set of simulated experiments using real data from the financial market. We have selected the thirty four assets from the Nasdaq 100 Index that present at least fifteen years of price history from January, 1995 to December, 2009. These assets are listed on Table 2.

Using this set of assets, we have performed many simulation experiments using a market simulator called AgEx [5]. We have tested COAST societies trading in the exchange, using daily quotes, where each coach could make one order a day. As discussed in Sect. 2.3, 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 [5].

### 4.1 Experimental setup

In order to analyze the raised questions above, we have executed many simulation experiments using four different COAST societies, and changing the time interval between two negotiation processes. In Table 3, these societies and the time interval between negotiation processes are presented.

The fifteen-year evaluation period was divided in five periods of three years, as shown in Table 4. We have also computed the performance average of the five trienniums. In most periods, as well as the average, the ***Very frequent negotiations*** and ***Frequent negotiations*** societies are the best in return, as it may be seen in Fig. 8. On the other hand, as shown in Fig. 9, there is no significant differences between the four societies regarding the risk. However, if we analyze jointly both the risk and return values through the Sharpe index, the two above mentioned societies also presented the best performance, as presented in Fig. 10. Hence, these two societies were selected to be compared with a theoretical portfolio specially created by us for analytical purposes.

Our first idea of direct comparison was to use the Nasdaq 100 (N100) index. However, a comparison among COAST performance and Nasdaq 100 (N100) index is biased because they do not deal with the same assets. In fact, N100 index composition changes all the time and many assets have been included or excluded along the fifteen years of the evaluation period, i.e. from 1996 to 2010. Due to these facts, we have created a theoretical portfolio called N34, which is composed of the thirty-four assets used by COAST societies. Since 2005, Nasdaq has not published the relative weights of N100 for each asset. Therefore, we defined N34 relative weights according to previous relative weight discrimination. Moreover, we normalized these weights to use only the chosen assets using Eq. (21), where $p_i$ is the asset

**Table 3** COAST societies and their time interval between negotiation processes. The time interval is expressed in coach's cycles (see Sect. 3.2.2)

| Society name | Time interval |
|---|---|
| No negotiations | $\infty$ |
| Rare negotiations | 50 |
| Frequent negotiations | 25 |
| Very frequent negotiations | 3 |

**Table 4** The five trienniums of evaluation period

| Trienniums | Start date | End date |
|---|---|---|
| 1o. triennium | Jan 01 2008 | Dec 31 2010 |
| 2o. triennium | Jan 01 2005 | Dec 31 2007 |
| 3o. triennium | Jan 01 2002 | Dec 31 2004 |
| 4o. triennium | Jan 01 1999 | Dec 31 2001 |
| 5o. triennium | Jan 01 1996 | Dec 31 1999 |

**Fig. 8** Daily return obtained by four COAST societies

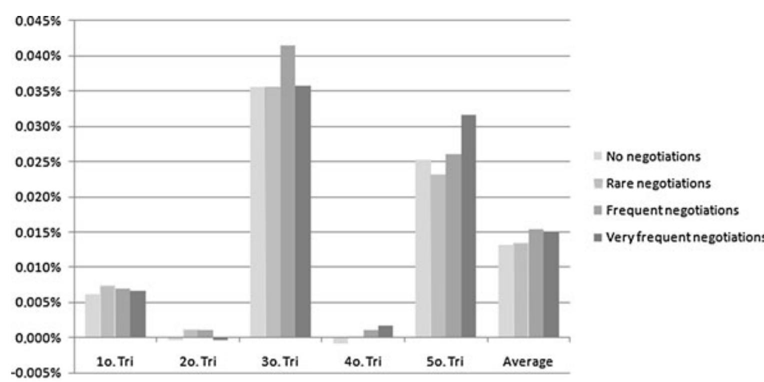
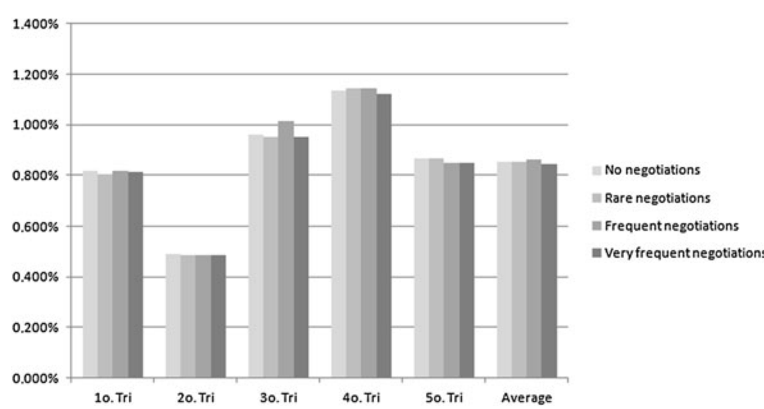



**Fig. 9** Risk presented by four COAST societies



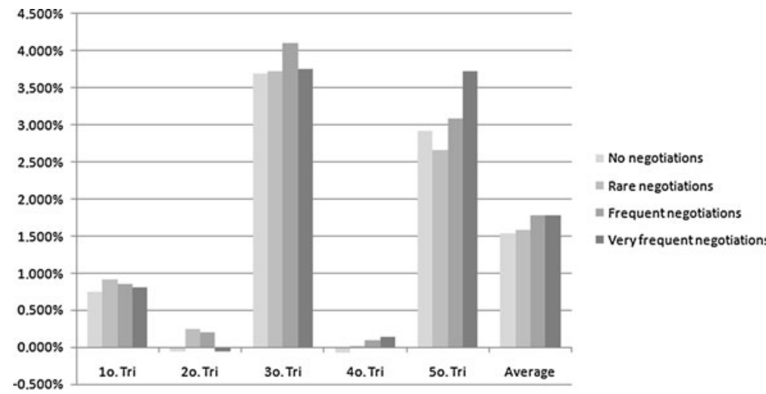



**Fig. 10** Sharpe index presented by four COAST societies

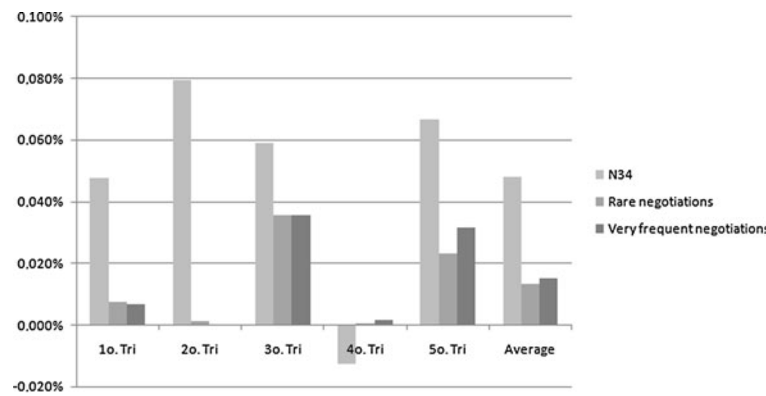



**Fig. 11** Daily return obtained by two COAST societies and N34 agent

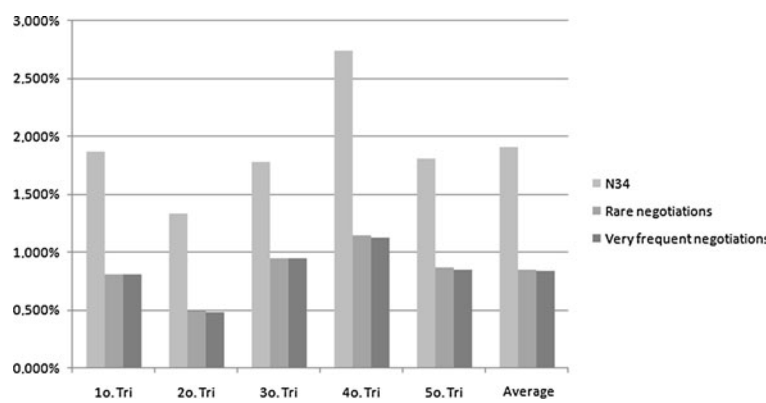



**Fig. 12** Risk presented by two COAST societies and N34 agent

weight in N34 theoretical index and $w_i$ is the original weight in N100.

$$p_i = \frac{w_i}{\sum_{j \in N34} w_j} * 100\ \% \qquad (21)$$

We have used these weights to define a trader agent, using AgEx, which buys and holds a set of shares according to the specified weights. This agent, also called N34, acted in the same simulated evaluation period of five trienniums. We compared N34 agent performance to ***Very frequent negotiations*** and ***Rare negotiations*** COAST societies. N34 presented a better performance in return, as may be seen in Fig. 11, but it was the worst according to the risk criterion, as presented in Fig. 12. Finally, Fig. 13 presents the performance of N34 agent and both COAST societies using the Sharpe Index.

## 4.2 Analysis of results

The comparison among the four COAST societies in return (Fig. 8) and the Sharpe index (Fig. 10) shows that frequent negotiations, i.e. frequent reallocation of resources, may bring higher performance, according to these two criteria. The ***Very frequent negotiations*** society achieves the best return and best Sharpe index in the average of the five trienniums. Regarding the risk criteria, the ***Very frequent negotiations*** society presents the highest risk in the fifth triennium, but it also presents the lowest risk in the

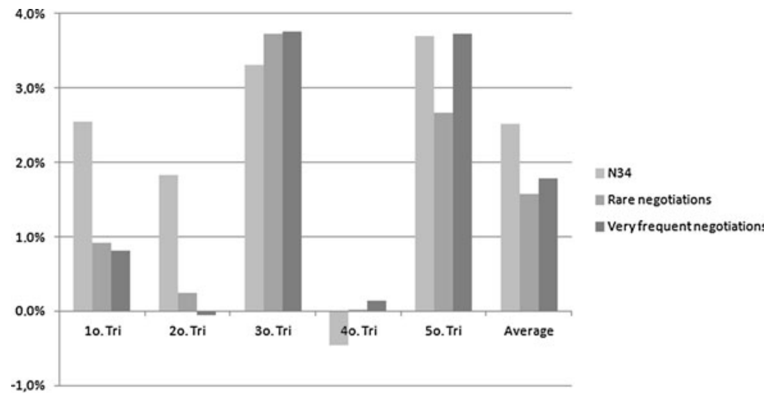



**Fig. 13** Sharpe index presented by two COAST societies and N34 agent

other four trienniums. These facts make us believe that it may be possible to pursue better results with more effective negotiations. By more effective, we mean negotiations with more impact in resource reallocations. Currently, only those coaches with bad performance may transfer resources, and at each negotiation process they may transfer just the money that is not already invested in assets. Additionally, coaches with neutral performance do not lose resources. When negotiation processes are made more frequently, there is a faster transfer to the best coach. The ***Very frequent negotiations*** society performs negotiations at three cycles interval, as stated in Table 3. In order to make more effective reallocations, we need to change the amount of resources that can be transferred at the end of a negotiation process. We intend to do that in the future.

On the other hand, analyzing the results in Figs. 11 and 12, it is possible to verify that ***Very frequent negotiations*** and ***Rare negotiations*** COAST societies overcome the N34 agent in the risk criterion, but the returns achieved by COAST societies are systematically lower than N34. This relatively bad performance in the return criterion is due to the fact that COAST achieves a best result in risk criterion at the cost of a poor performance in return. Moreover, if we analyze the performance measured through Sharpe Index as shown in Fig. 13, it is possible to realize that COAST societies perform better in two trienniums, while N34 performs better in three trienniums; consequently, the later also has a better performance in the average of the five trienniums. However, these results are quite interesting if we consider investors whose main goal is to fix a maximum risk limit. As mentioned before, to our knowledge the COAST architecture is the only automated asset management system capable of dealing with these kinds of investor profiles, when considering the non-existence of any risk free asset. Furthermore, we believe it is possible to improve COAST performance with the addition of new advisor agents, rather than using only the current five advisors that are based on common technical analysis indexes.

## 5 Conclusions and further work

In this paper, we presented the multiagent COAST architecture, its agents, and a new negotiation mechanism especially designed for this architecture. This architecture was implemented and tested in several simulation experiments. These experiments were presented and analyzed. COAST societies performed well in risk criterion and those with more frequent negotiations performed better than societies with less frequent negotiations according to return criterion. This result indicates that it may be possible to achieve a better performance in the return criterion with more effective resource reallocation among coaches, i.e., more resources being transferred at the end of each negotiation process.

The main contributions of this study are the following: its capacity to *represent different investor's preferences* with pre-defined acceptable limits of risk and/or return that guides the decisions of the whole automated system; the *exploitation of competitive strategies* within the COAST architecture that facilitates the use of well-known trading strategies as advisors agent strategies; and finally, the fact that it does not need to assume the existence of any risk free asset is also an significant advantage.

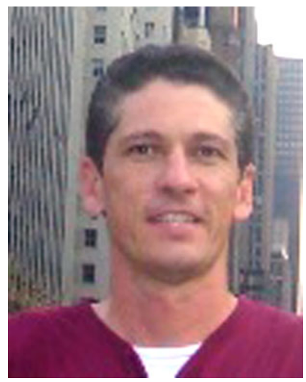
In future work, we intend to test COAST architecture with more trading strategies, using a wider evaluation period and number of assets. We believe that a significant evolution would be a formal modeling of expectations, which are very important in economic reasoning. Finally, despite the fact that the architecture was designed for automated asset management, we believe that it can be adapted to other complex multiagent environments.

**Acknowledgements** 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 N34 agent.

## References

1. Aldridge I (2009) High-frequency trading: a practical guide to algorithmic strategies and trading systems. Wiley, New York

2. Araujo CHD, Castro PAL (2010) Towards automated trading based on fundamentalist and technical data. In: Proceedings of 20th SBIA, São Bernado do Campo, Brazil. LNAI. Springer, Berlin, pp 704–715
3. Cañete A, Constanzo J, Salinas L (2008) Kernel price pattern trading. Appl Intell 29:152–156. doi:10.1007/s10489-007-0054-2
4. Castro PA, Sichman JS (2007) Towards cooperation among competitive trader agents. In: Proceedings of 9th ICEIS, Funchal, Portugal, pp 138–143
5. Castro PA, Sichman JS (2009) Agex: a financial market simulation tool for software agents. In: Aalst W, Mylopoulos J, Sadeh NM, Shaw MJ, Szyperski C, Filipe J, Cordeiro J (eds) LNBIP, vol 24. Springer, Berlin, pp 704–715
6. Decker K, Pannu A, Sycara K, Williamson M (1997) Designing behaviors for information agents. In: Johnson WL, Hayes-Roth B (eds) Proceedings of the first international conference on autonomous agents (Agents'97). ACM, New York, pp 404–412
7. Dougherty ER (1990) Probability and statistics for the engineering, computing, and physical sciences. Prentice Hall, Englewood Cliffs
8. Durbin M (2010) All about high-frequency trading. McGraw-Hill, New York
9. Fabio Bellifemine GC, Greenwood D (2007) Developing multi-agent systems with JADE. Wiley, Chichester
10. Farmer JD, Foley D (2009) The economy needs agent-based modelling. Nature 460:685–686
11. Feng X, Jo C-H (2003) Agent-based stock trading. In: Proceedings of the ISCA CATA-2003, Honolulu, USA
12. Francis JC (1983) Management of investments. McGraw-Hill, New York
13. Garcia MEF, Marin EA, Garcia RQ (2010) Improving return using risk-return adjustment and incremental training in technical trading rules with gaps. Appl Intell 33(2):93–106
14. Kearns M, Ortiz L (2003) The Penn-Lehman automated trading project. IEEE Intell Syst 18(6):22–31
15. Kendall G, Su Y (2003) 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
16. Kendall G, Su Y (2005) A particle swarm optimisation approach in the construction of optimal risky portfolios. In: Proceedings of the 23rd IASTED international multi-conference artificial intelligence and applications, Innsbruck, Austria, pp 140–145
17. Lee S, Ahn J, Oh K, Kim T (2010) Using rough set to support investment strategies of real-time trading in futures market. Appl Intell 32:364–377. doi:10.1007/s10489-008-0150-y
18. Markowitz HM (1952) Portfolio selection. J Finance 7(1):77–91
19. Mochón A, Quintana D, Sáez Y, Isasi P (2008) Soft computing techniques applied to finance. Appl Intell 29:111–115. doi:10.1007/s10489-007-0051-5
20. Pedrycz W, Gomide F (1998) An introduction to fuzzy sets: analysis and design. MIT Press, Cambridge
21. Rosenschein JS, Zlotkin G (1994) Rules of encounter: designing conventions for automated negotiation among computers. MIT Press, Boston
22. Russell S, Norvig P (2003) Artificial intelligence. A modern approach, 2nd edn. Prentice Hall, Englewood Cliffs
23. Sharpe WF (1994) The Sharpe ratio. J Portf Manag 13(3):227–286
24. Sherstov A, Stone P (2004) Three automated stock-trading agents: a comparative study. In: Proceedings of the agent mediated electronic commerce (AMEC) workshop—AAMAS 2004, New York
25. Shoham Y, Leyton-Brown K (2009) Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, New York
26. Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans Comput 29(12):1104–1113
27. Sortino F A, Forsey HJ (1996) On the use and misuse of downside risk. J Portf Manag 22(1):35–42
28. Tan A, Quek C, Yow K (2008) Maximizing winning trades using a novel rspop fuzzy neural network intelligent stock trading system. Appl Intell 29:116–128. doi:10.1007/s10489-007-0055-1
29. Weiss G (1999) Multiagent systems: a modern approach to distributed artificial intelligence. MIT Press, Cambridge
30. Wooldridge M (2002) An introduction to multiagent systems. Wiley, London
31. Yuan Luo KL, Davis DN (2002) A multi-agent decision support system for stock trading. IEEE Netw 16(1):20–27



**Paulo André Lima de Castro** is an Assistant Professor at Instituto Tecnológico de Aeronáutica (ITA), where he obtained his B.Sc. in Computer Engineering in 1997. He has achieved his M.Sc. and Ph.D. (2009) degrees at Escola Politécnica da Universidade de São Paulo. He has experience in software development and researches mainly in multiagent systems, artificial intelligence and automated trading.

**Jaime Simão Sichman** is an Associate Professor at the Escola Politécnica da Universidade de São Paulo, Brazil, where he has obtained both his B.Sc. and M.Sc. degrees in Electrical Engineering. He was one of the first students to get a European label for his Ph.D. studies, jointly developed at the Institut National Polytechnique de Grenoble (INPG), France and at the Istituto di Psicologia del CNR, Rome, Italy. More recently, he has spent a brief post doctoral period at the University of Utrecht, in the Netherlands. His main focus of research are multiagent systems, more particularly social and organisational reasoning, interoperability, reputation, and multiagent-based simulation. He has advised and co-advised 11 Ph.D. and 9 M.Sc. students. He was one of the founders of the Multi-Agent-Based Simulation (MABS) and of the Coordination, Organization, Institutions and Norms in Agent Systems (COIN) international workshop series. He has published more than 160 articles in national and international journals and conferences. He was one of the Program Co- Chair of AAMAS 2009 and of SBIA/IBERAMIA 2006. He is a member of the editorial board of the Journal of Artificial Societies and Social Simulation, Mediterranean Journal of Artificial Intelligence, Computación y Sistemas, Inteligencia Artificial and Knowledge Engineering Review. Currently, he is the director of the Centro de Computação Eletrônica da Universidade de São Paulo.