# Improving Classifier Agents with Order Book Information

Philippe Mathieu and Matthis Gaciarz

Laboratoire d'Informatique Fondamentale de Lille
LIFL UMR 8022 CNRS
Lille1 University
59655 Villeneuve d'Ascq, France
philippe.mathieu@lifl.fr

**Abstract.** In the study of financial phenomena, agent-based artificial markets are efficient tools for testing economic assumptions of market regulation. While it is easy to populate these virtual worlds with "basic" chartist agents using price history (increase or decrease of prices, moving averages...), it is nevertheless necessary, in order to study rationality phenomena and influence between agents, to add some kind of learning agents. Several authors have of course already been interested in adaptive techniques but they mainly take into account price history. But prices are only consequences of orders and therefore reasoning about orders provides a head start in the deductive process. In this paper we show how to take into account all of the information about the market, including how to leverage the information from order books such as the best limits, size of bid-ask spread or cash at hand waiting to accommodate more effectively to market offerings. Like B. Arthur we focus here on the use of LCS agents.

## 1 Introduction

For a few years, advances in computer science have provided powerful tools to study complex economic systems. Individual-based approaches provide a high level of detail and various advantages. It is now possible to test regulation systems, or the influence of new policies on an individual scale, and not only group scale. Among these economic systems, artificial stock markets now offer a credible alternative to mathematical finance and financial econometrics. With these tools, macroscopic phenomena become the consequence of microscopic phenomena, because decision processes and actions are made by the individuals. Thanks to these advantages, multi-agent systems show all their potential.

### 1.1 A Multi-agent Artificial Stock Market

In the literature many artificial markets use an equation based price fixing mechanism [1] [2]. In these markets, the agents send bid or ask signals on the market and the market fixes prices according to the total number of each signal. This

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 204–215, 2013.

method is different from how real marketplaces work. Moreover it is not powerful enough to allow endowing agents with powerful intelligent strategies, reasoning on the type, the price and the quantity of each order. This level of detail is required to test the consequences of regulation rules on individuals, the social well-being of societies [3] or the influence of speculators in the trader population.

The artificial market platform ATOM [4], is a tool that clones the main features of the Euronext-NYSE stock exchange microstructure. One of these features is the double-auction order-book system : for each asset, the ask part contains orders with prices in increasing order and the bid part contains orders with prices decreasing order. Like Euronext, ATOM aims at matching orders sent by virtual traders to determine quotations and prices. Orders are sent by agents that all have their own strategy.

With its realistic design, ATOM allows agents to access to a lot of data. There is the price history (Fig. 1), that is the only information used by equational systems, but also the order history and the order book (Fig. 2). The order book shows the state of the bid and the ask at time $t$, and contains information about the orders. Thus ATOM allows agents to take advantage of much information to develop their strategies.

| Price | ... | 110.8 | 110.9 | 111.0 | 110.9 |
|---|---|---|---|---|---|
| Quantity | ... | 5 | 2 | 7 | 8 |

**Fig. 1.** History of fixed prices, with the traded quantities

| Dir | Order | Issuer | Quantity | Price |
|---|---|---|---|---|
| | ... | ... | ... | ... |
| | o1 | Ag2 | 8 | 111.5 |
| | o2 | Ag1 | 10 | 111.1 |
| Ask | o4 | Ag3 | 3 | **111.0** |
| | | | bid-ask spread ↕ | |
| Bid | o10 | Ag4 | 2 | **110.8** |
| | o3 | Ag1 | 6 | 110.6 |
| | ... | ... | ... | ... |

**Fig. 2.** Order book for one particular asset. The best prices ar shown in bold font. The interval between the best offers is called bid-ask spread.

For many years, economic theories have asserted that it is important to take the information of the order books into account [5] [6] [7], but no one has highlighted it experimentally so far until now.

The goal of this paper is to show that it is possible to design agent behaviours that take advantage of this information to build more rational behaviours.

### 1.2 Learning Agents

Like multi-agent systems, machine learning techniques have been developed over the last decades. The main learning categories (supervised, unsupervised, reinforcement learning) are based on various algorithms [8].

Many of them have been used on artificial markets, with more or less success [9] [10]. Nevertheless, all these techniques have a lack of explanation: once the learning has been achieved, it is difficult to understand why the decisions are taken, or why specific behaviours have been triggered.

Like B Arthur, we choose to support this work with a learning technique not often used in literature, but more suitable for explanatory models: classifier systems. Other techniques, like the ones we mentioned above, would work too. The purpose of our work is not to compare these, but to show how to design an adaptive and explanatory model that uses all the market information to provide sophisticated and varied behaviours. A Learning Classifier Systems (LCS, [11]) contains a set of binary rules. The quality of a rule, called fitness, is modified at each step by a reinforcement learning technique. Moreover, a genetic algorithm periodically performs a natural selection, based on classical mutation and selection mechanisms.

One of the first artificial markets, the SF-ASM (Santa Fe Artificial Stock Market, [1] [2]), already used LCS for the agent reasoning, but it was an equational market, and its agents can only take the price history in account.

### 1.3 Classifier Systems

Before tackling the whole complexity of a market, let us describe how a usual classifier system works [1], illustrating it within the framework of an equational price-fixing market, using agents that only decide to buy or sell.

An LCS uses a set of conditions on market state called market descriptors. These descriptors can be seen as the "sensors" used by the LCS to perceive the market. With these, it is possible to analyse the market and build a binary sequence whose length is equal to the number of descriptors used. Figure 3 shows an example of a descriptor set. The first descriptor Ex1 is satisfied if the current price is higher than the previous price. Ex2 is satisfied if the current price is higher than the average of the last five prices. Ex3 is satisfied if the current price is less than 100.

Once these descriptors have been chosen, each LCS is endowed with set of rules (or classifiers) from a triplet (state, score, action):

– State $S$ of a rule is a trinary sequence that determines whether this rule can be activated considering the current situation. The trits (**tr**inary dig**its**) can be set to 1, 0 or #. In the sequence, each trit matches a descriptor. If the value of a trit is 1, the descriptor has to be satisfied to activate the rule. If

| | |
|---|---|
| Ex1 | $p_t > p_{t-1}$ |
| Ex2 | $p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$ |
| Ex3 | $p_t < 100$ |

**Fig. 3.** Descriptor examples that can be used by a LCS

it is 0, the descriptor has to be unsatisfied to activate the rule. The sign # is a wildcard, meaning that the descriptor is not taken into account in the rule's activation process.
- the fitness score $F$ represents the trust allotted to this rule, according to its previous forecast successes
- the action $A$ represents the action to perform when the rule is activated

An LCS can have up to $3^n$ rules to reason on, $n$ being the number of descriptors. Of course, it can have fewer.

Figure 4 is an example of a five-rule classifier system using three descriptors. for instance, the first rule R1 can be activated if the current situation of the market matches with state 010 or with state 110. This rule allows the LCS to select a bid order, and its score is 5.

| | state | action | score |
|---|---|---|---|
| R1 | #10 | buy | 5 |
| R2 | 1#0 | sell | 18 |
| R3 | 00# | buy | 12 |
| R4 | 110 | buy | 4 |
| R5 | #11 | sell | 9 |

**Fig. 4.** Example of a 5 size classifier system

An LCS works as follows : each time it has to take a decision, it selects a rule among the activable ones, with a random mechanism in which the probability for each rule to be chosen is proportional to its score.

The LCS agent performs the action of the selected rule (it sends a bid or a ask signal). During the next LCS activation, the score of each activable rule is updated. It is increased or decreased according to the correctness of the forecast produced by the rule.

A genetic algorithm is periodically applied on the set of rules, using the score of each rule to perform a natural selection in the classifier system. If a score does not reach a threshold, the rule is eliminated. The best rules are crossed and mutated to regenerate the population. Combining the continuous evaluation of the rules and the genetic algorithm allows the LCS to perform an effective learning routine. It is possible that some rules cannot be activated because of contradictory descriptors. To avoid this and keep the classifier system size constant, the rules not activated for a long time are eliminated by the genetic algorithm.

## 2 Proposal

Setting up LCS in each agent in an order-driven market raises several problems. The previous LCS only chooses directions, but in an order-driven market, other information is necessary for an agent to build an order. The most common being the LimitOrder in which the agent has to generate a quantity and a price.

Moreover, it is important to take advantage of pending orders. Finally, setting up an LCS requires to clearly define temporal references for the descriptors that can be interpreted in various ways.

When an agent sends an order to the market, the direction of this order is determined by the activated rule. On a real market, and on ATOM, it is also necessary to set a price limit and a quantity for this order. To do this, we propose to use a simple policy, made up of a price-setting strategy and a quantity-setting strategy. For each of these two strategies, there are various possibilities:

Price-setting strategies :

- Setting the price so that the order sent is placed at the top of the order book:
  $P_{Bid} = P_{BestBidOrder} + \varepsilon$
  $P_{Ask} = P_{BestAskOrder} - \varepsilon$
- Producting an order that will be immediately executed (at least partially), setting a price equal to the best rival order's price :
  $P_{Bid} = P_{BestAskOrder}$
  $P_{Ask} = P_{BestBidOrder}$

To set the quantity, various strategies are possible too.

Quantity setting strategies:

- Constant quantity : $Q = k_c$
- Quantity proportional to the chosen rule's score : $Q = k_p F$

Our experiments have shown that the policy which give the best result is the policy that sends orders to the top of the order book, with a constant quantity. The next results will be based on this policy.

In an asynchronous order-driven market, other information is available for the agent, based on the past orders and the current order book. We propose to use this information in the classifier system.

### 2.1 Improve Agents Behaviour Learning from Orders

As equational systems are the most widespread ones, the common reasoning in finance is to perform a technical analysis of the price history to deduce a future increase or decrease. It is the case of "chartist" agents, that search for particular shapes in the price series. In order to show the contribution of the order book to agent reasoning, in this paper we propose to use a price-based LCS agent and to improve it adding new order-based descriptors. Then with a set of experiments we show that these descriptors provide relevant and useful information for decision-making.

The agent used as a reference in this work uses the price-based criteria introduced in Fig. 5. These are interesting because they allow to take into account price variations both in the short and the long term, on three simple criteria : price increase compared to an older price (descriptor 1), compared to the average of the last $n$ prices (descriptors 2 to 4), or compared to the midrange of the last $n$ prices (descriptors 5 and 6).

| | |
|---|---|
| 1 | $p_t > p_{t-1}$ |
| 2 | $p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$ |
| 3 | $p_t > 1/10 \times \sum_{i=t-1}^{t-10} p_i$ |
| 4 | $p_t > 1/100 \times \sum_{i=t-1}^{t-100} p_i$ |
| 5 | $p_t > 1/2[Minp_i + Maxp_i]_{i \in [t-1,t-10]}$ |
| 6 | $p_t > 1/2[Minp_i + Maxp_i]_{i \in [t-1,t-100]}$ |

**Fig. 5.** Technical analysis descriptors used by our LCS agent

One of the main data contained in the order book is the gap between the best bid and the best ask, called bid-asked spread (Fig. 2). This is a common measure of the liquidity of a market, because the wider the spread is, the more important the consequences are in case of mistake in reasoning (intuitively, the cost implied by selling an asset and buying it again is higher).

In order to efficiently take advantage of the information in the order book, we propose to give our LCS agents two new kinds of descriptors. The first ones take into account the value and the evolution of the bid-ask spread, and the second ones take into account the imbalance between bid and ask. The descriptors we introduce here are evaluated in the results section.

**The Bid-Ask Spread.** The most intuitive approach is probably to reason on the value of the spread (descriptor 7, Fig. 6). But it does not take the scale of the values into account, that is why the proportion is a more relevant parameter. We prefer to use the ratio $r = \frac{bestP_{Ask}}{bestP_{Bid}}$ (descriptor 8).

One can, for example, compare the current value of this ratio to a previous value (descriptor 9) to determine whether the current value of the bid-ask spread is rather high or rather low. We can also use the average of the $k_{10}$ previous values of $r$, or the midrange of the $k_{11}$ previous values.

| | |
|---|---|
| 7 | $bestP_{Ask} - bestP_{Bid} < k_7$ |
| 8 | $r_t < k_8$ |
| 9 | $r_t > r_{t-k_9}$ |
| 10 | $r_t > 1/k_{10} \times \sum_{i=t-1}^{t-k_{10}} r_i$ |
| 11 | $r_t > 1/2[Minr_i + Maxr_i]_{i \in [t-1,t-k_{11}]}$ |

**Fig. 6.** Descriptors based on the variation of the bid-ask Spread and the $r = \frac{bestP_{Ask}}{bestP_{Bid}}$ ratio

**Imbalance between Bid and Ask.** The relative size of the two parts of the order book (in quantity of assets to trade) is useful information, because it can reveal an imbalance between bid and ask (Fig. 7). Either way, this imbalance may indicate a future variation of the price, and the agent can take advantage of it. Descriptors 12 and 13 check if there are orders in each part of the order book, and descriptor 14 reasons on the ratio $\frac{nbAsks}{nbBids}$.

The relative size of each part of the order book is not the only way to evaluate imbalances between bid and ask. Indeed, if 10 assets are sold at the same price $p_0$, the ask is better than if 1 asset is sold at $p_0$ and the 9 other have a higher price, but the ratio $\frac{nbAsks}{nbBids}$ does not change. To take the price into account, we propose this measure of bid and ask :

$$ask = \sum_{order \in AskOrderBook} \frac{Quantity(order)}{Price(order) - bidAskMid}$$
$$bid = \sum_{order \in BidOrderBook} \frac{Quantity(order)}{bidAskMid - Price(order)}$$

with:

$$bidAskMid = \frac{bestP_{ask} + bestP_{bid}}{2}$$

$bidAskMid$ is the average of the best selling price and the best buying price. By dividing the quantity of each order by the difference between its price and $bidAskMid$, we take in account the fact that some orders have a limit price too low or too high to constitute an attractive bid or ask. We are interested in the ratio $q = \frac{ask}{bid}$ (descriptor 15) and its variations (descriptors 16 to 18).

| | |
|---|---|
| 12 | No bid in the order book |
| 13 | No ask in the order book |
| 14 | $\frac{nbVentes}{nbAchats} > k_{14}$ |
| 15 | $q_t > k_{15}$ |
| 16 | $q_t > q_{t-k_{16}}$ |
| 17 | $q_t > 1/k_{17} \times \sum_{i=t-1}^{t-k_{17}} q_i$ |
| 18 | $q_t > 1/2[Minq_i + Maxq_i]_{i \in [t-1, t-k_{18}]}$ |

**Fig. 7.** Descriptors based on the size of each part of the order book and on the $q = \frac{ask}{bid}$ ratio

The descriptors on Fig. 6 and 7 are only examples of order-book-based descriptors, but it is of course possible to design and test others. Descriptors that use a constant $k$ were implemented several times, with various values of this constant.

## 3 Methodology

Evaluating agent behaviours is a difficult task. Firstly because like for voting systems it is always possible to favour a particular agent. Secondly because an agent is rarely performant in itself, but relative to its competitors and to its environment.

In order to compare agents, it is necessary to define how the performance of an agent is measured. Two classics are possible in our case: the cash owned by an agent, and the amount of this cash with the estimated value of its portfolio. This sum is called wealth.

$$wealth = cash + \sum_{i=1}^{i<assets} price_i \times nbAssets_i$$

Even if this measure can be discussed since it is an approximation, it allows to take into account all the possessions of the agent. That is why we use this measure for our work.

Now that we have chosen a measure, there are many ways to evaluate agent performance in a group. Two main kinds of evaluation coexist [12]: evaluation in which the $n$ agents evolve in the same environment and are competing with one another, and evaluations in which the $n$ agents are ranked relatively to the same set of competitors.

The ecological competition is a selection method inspired by biology and natural selection ( [13], [14]). Several families coevolve like animal or vegetal species sharing an environment. Like in nature, their populations vary according to time, so that the population of the best families increases, and the population of the worst families decreases.

In our case, a competition is a series of simulations where the total number of agents is constant. Each family starts with an identical number of agents. After each simulation, called "generation", the population of each family is evaluated according to its profits. In order to keep a constant total population, we have to apply a proportionality rule on the score of each family.

The score of a family is the total profits of its members during the simulation. But the individual profits of an agent can be negative, therefore the total profits of a family can be negative too, and applying a proportionality rule requires having positive values. To solve this problem, we propose to subtract from the profits of each agent $a$ the profits of the worst agent in the simulation $w$.

$$profits_a \geq profits_w \Rightarrow profits_a - profits_w \geq 0$$

The modified profits of each agent is thus positive, as well as the total profits of each family. It is then possible to apply a proportionality rule to compute the new population of each family.

The total profits of family $f$ is the sum of the modified wealth of all the agents of this family.

$$totProfits(f) = \sum_{a \in f} (profits_a - profits_w)$$

The competition has a constant total population. At the end of each generation, the population of a family is proportional to its total profits during the previous simulation.

$$pop(f) = \frac{totProfits(f)}{\sum_{i \in families} totProfits(i)} totalPop$$

The population of an agent family at the end of a competition represents how well this kind of agent fits in a particular environment (the agent families in competition), and its effectiveness in this environment.

It is difficult to maintain that an agent is better than another in itself. However, if an agent obtains better results than another in several ecological competitions that are different enough to provide various environments, one can

postulate that this agent is globally better. This is this method that we used to evaluate our agents.

## 4 Results

Many simulations have been run to obtain the results shown in this section. To conduct them, we used a specific protocol.

### 4.1 Experimental Protocol

In order to highlight the contribution of an order-book-based learning, in this paper we have decided to compare various kinds of LCS agents. One of these only uses a price-based LCS with the descriptors of figure 5). We use these agents as basic agents for our comparisons. Our goal is to overtake these.

We compare this basic agent type to various kinds of prices and order-based LCS agents, that use the same price-based descriptors as the basic agent, but also one or several order-based descriptors, from figures 6 and 7).

In order to generate varied situations, the competitions are populated with various agent families:

- chartist agents (moving average, RSI, momentum, variation, indicators and mixed moving average): these agents use simple conditions on the prices to forecast their future variation and decide to buy or sell
- periodic agents : periodically buy and sell
- Zero Intelligence Traders (ZIT, [15]) : these agents send orders in a random direction with a random limit price

All the learning-agent families are used in several ecological competitions, that are the same for every tested family. For each competition, the final populations of these families are compared to determine the best kind of agent for a particular environment. For each family, this operation is repeated in about ten competitions, five times per competition. Fifty generations per competition are enough to allow the population to stabilise in most cases. Each generation is equivalent to one day in which 2000 decisions are taken by each agent, enough to allow the learning agents to adapt.

### 4.2 Results Obtained

Figure 8 shows the average population of several agent families for various competitions. In each family, the agents have the same descriptors as the basic price-based agent, and we add to them a set of 1 to 5 order-based descriptors. We observe that several families obtain better results than the basic agent. The descriptor sets of the families presented in figure 8 are those that give best improvements. These use up to 3 new descriptors, that are instances of descriptors 10, 11, 16 and 17 (see Fig. 9). The agents that use these interesting descriptors (the other descriptors give little or no improvement) have a final population 100% greater than the basic agent. Therefore we can consider that LCS agents are really improved by learning with orders.

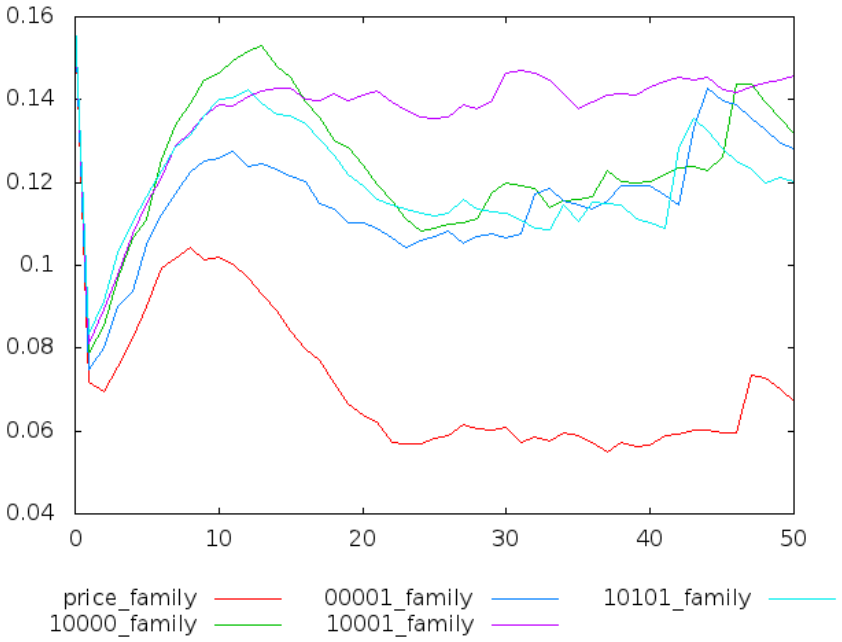


**Fig. 8.** Average part of the total population during generations, for the basic agent type (price_family) and various families using order-based descriptors. The binary sequence in the name of each family describes the additional descriptors used. Each bit refers to a descriptor in Fig. 9, in the same order.

| | $k$ | Descriptor's instance |
|---|---|---|
| 10 | 100 | $r_t > 1/100 \times \sum_{i=t-1}^{t-100} r_i$ |
| 11 | 100 | $r_t > 1/2[Minr_i + Maxr_i]_{i\in[t-1,t-100]}$ |
| 16 | 5 | $q_t > q_{t-5}$ |
| 17 | 10 | $q_t > 1/10 \times \sum_{i=t-1}^{t-10} q_i$ |
| 17 | 100 | $q_t > 1/100 \times \sum_{i=t-1}^{t-100} q_i$ |

**Fig. 9.** Order-based descriptors improving agents results

### 4.3 Adding Descriptors to an LCS Agent

The more information has an agent, the more likely it is to give an accurate forecast of the price variation. Nevertheless, adding a descriptor to an agent broadens its research scope, and makes the learning slower or even less efficient.

All the information does not have the same relevance. For example, we can suppose that descriptor $p_t > p_{t-1}$ is more relevant than descriptor $p_{t-100} > p_{t-101}$ in many cases. But the broadening of the search scope being the same whatever the added descriptor is, the information added by a descriptor has to be significant to offset this broadening. Moreover, if two descriptors provide

similar information, the value added by the second one is very low. An LCS has to use a limited descriptor set (from 6 to 9), and these have to bring significant and varied information.

## 5 Conclusion

To carry out realistic financial simulations, it is important to populate artificial markets with adaptive behaviour agents.

Until now, in the absence of software platforms based on orders that use a multi-agent approach, this kind of simulation was conducted with a basic equational price-fixing model. For example this is the case in the well-known SF-ASM. The ATOM platform, with its completeness and its high relevance to order-driven markets like Euronext-NYSE, allows to improve the learning ability of agents.

In this paper, after having detailed the various possibilities to reason on the orders and their consequences, we have shown how to develop classifier systems that take into account all the system information for each agent. In order to compare these agents, we have implemented an original adaptation of an ecological competition that allows us not only to measure the performance of an agent, but also its robustness to environment modifications. Thus, we have highlighted that an agent that studies pending orders is far more efficient than its counterpart which only reasons on the prices.

Further work has to be done in this area : varying the learning methods, the descriptors, or trying to recognise the individual behaviour of the agents. We consider that this paper is a one step in the development of efficient order-based learning agents.

## References

1. Arthur, W.B., Holland, J., LeBaron, B., Palmer, R., Tayler, P.: Asset pricing under endogenous expectations in an artificial stock market. In: The Economy as an Evolving Complex System II, pp. 15–44. Addison-Wesley, Reading (1997)
2. LeBaron, B.: Building the santa fe artificial stock market. Brandeis University internal report (2002)
3. Brandouy, O., Mathieu, P.: Efficient monitoring of financial orders with agent-based technologies. In: Demazeau, Y., Pechoucek, M., Corchado, J.M., Pérez, J.B. (eds.) Adv. on Prac. Appl. of Agents and Mult. Sys. AISC, vol. 88, pp. 277–286. Springer, Heidelberg (2011)
4. Mathieu, P., Brandouy, O.: A generic architecture for realistic simulations of complex financial dynamics. In: Demazeau, Y., Dignum, F., Corchado, J.M., Pérez, J.B. (eds.) Advances in PAAMS. AISC, vol. 70, pp. 185–197. Springer, Heidelberg (2010)
5. Belter, K.: Supply and information content of order book depth: The case of displayed and hidden depth (2007)
6. Cao, C., Hansch, O., Wang, X.: The informational content of an open limit order book. In: EFA 2004 Maastricht Meetings Paper (2003)

7. Kozhan, R., Salmon, M.: The information content of a limit order book: The case of an fx market. Journal of Financial Markets 15(1), 1–29 (2012)
8. Cornuéjols, A., Miclet, L., Kodratoff, Y.: Apprentissage Artificiel: Concepts et algorithmes. Eyrolles (2002)
9. Barbosa, R.P., Belo, O.: An agent task force for stock trading. In: Demazeau, Y., Pechoucek, M., Corchado, J.M., Pérez, J.B. (eds.) Adv. on Prac. Appl. of Agents and Mult. Sys. AISC, vol. 88, pp. 287–297. Springer, Heidelberg (2011)
10. Cao, L., Tay, F.: Application of support vector machines in financial time series forecasting. Omega: The International Journal of Management Science (29), 309–317 (2001)
11. Booker, L., Golberg, D., Holland, J.: Classifier systems and genetic algorithms. Artificial Intelligence 40(1-3), 235–282 (1989)
12. Beaufils, B., Mathieu, P.: Cheating is not playing: Methodological issues of computational game theory. In: ECAI 2006, pp. 185–189. IOS Press (2006)
13. Lotka, A.: Elements of Physical Biology. Williams and Wilkins Company (1925)
14. Volterra, V.: Variations and fluctuations of the number of individuals in animal species living together. ICES Journal of Marine Science 3(1), 3–51 (1928)
15. Gode, D.K., Sunder, S.: Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. Journal of Political Economy 101, 119–137 (1993)