Using LCS to Exploit Order Book Data in Artificial Markets

Philippe Mathieu and Yann Secq(✉)

Laboratoire d'Informatique Fondamentale de Lille (UMR CNRS 8022),
Université Lille 1, Villeneuve-d'Ascq, France
{philippe.mathieu,yann.secq}@univ-lille1.fr
http://www.lifl.fr/SMAC

Abstract. In the study of financial phenomena, multi-agent market order-driven simulators are tools that can effectively test different economic assumptions. Many studies have focused on the analysis of adaptive learning agents carrying on prices. But the prices are a consequence of the matching orders. Reasoning about orders should help to anticipate future prices.

While it is easy to populate these virtual worlds with agents analyzing "simple" prices shapes (rising or falling, moving averages, ...), it is nevertheless necessary to study the phenomena of rationality and influence between agents, which requires the use of adaptive agents that can learn from their environment. Several authors have obviously already used adaptive techniques but mainly by taking into account prices historical. But prices are only consequences of orders, thus reasoning about orders should provide a step ahead in the deductive process.

In this article, we show how to leverage information from the order books such as the best limits, the bid-ask spread or waiting cash to adapt more effectively to market offerings. Like B. Arthur, we use learning classifier systems and show how to adapt them to a multi-agent system.

Keywords: Agent based computational economics · Artificial stock market · Market microstructure · Learning classifier systems · Multi-agent simulation

1 Introduction

In recent years, advances in computer research have provided powerful tools for studying complex economic systems. Individual-based approaches, for their benefits and the level of detail they provide, are becoming increasingly popular within industries and even for policy makers. It is now possible to simulate complex economic systems to study the effects of new regulations, or the influence of new policies at the individual and not only at the group level.

Among these economic systems, artificial financial markets now offer a credible alternative to mathematical finance and econometrical finance. Thanks to multi-agent systems, decision-making as the actions taken can be individualized, macroscopic phenomena becomes consequences of microscopic interactions.

R. Kowalczyk et al. (Eds.): TCCI XV, LNCS 8670, pp. 69–88, 2014.
DOI: 10.1007/978-3-662-44750-5_4

1.1 An Artificiel Agent-Based Stock Market

One can found many artificial markets in the literature, but they are mostly built at the macroscopic level, with prices defined through equations [1,15]. Agents reason on price alone and send simple signals to the market to buy or sell an asset. The market contains no order book and sets the price only on the differential between asks and bids signals. This approach ignores the complexity of real markets and is insufficient to test sophisticated behavioural assumptions like reasoning on types of orders, prices or quantities. Thus, to evaluate possible consequences of regulatory rules on individuals, or to study societies social welfares [7] or even speculators influences within traders population, a granularity at the individual is required.

The ATOM multi-agent platform [8] implements an order-driven market that reproduce the main features of the microstructure of marketplaces like *EuroNEXT-NYSE* , including its system of double book of orders for each asset. In ATOM, agents rely on their own strategies to send orders on different assets. ATOM is built on classic agent design-patterns [17] to allow the enforcement of equity between agents and to conduct experimentations at several scales: from an intra-day level (intraday) by reasoning on each fixed price to multi-day scale (extraday) by reasoning only on the closing prices. Although ATOM supports multi-assets negotiation, this article focuses on the reasoning on a single order book. Unlike macroscopic market models, agents do not emit a simple signal to buy or sell but can send real orders consistent with those allowed on *EuroNEXT*. However, in this paper, we limit ourselves to the two most common types: limit and market orders.

In this paper, we use only the best known and most widely used order, the `LimitOrder` which is defined by:

- the order issuer,
- the asset's name to be exchanged,
- the desired direction (bid or sell),
- the number of asset to be exchanged,
- the price limit (i.e. the maximum accepted price for a purchase order and the minimum accepted price for a sell order).

By its relevance to the real market mechanisms, ATOM provides access to numerous data. Among these, there is the price history (Table 2), but also and especially the orders history (Table 1). Double order books that reflect the state of the offer at time t is also accessible, with information on all orders its contains (Table 3).

These information are significant and allow a measure definition of how easily an agent will find a counterpart to his orders (so called *market liquidity*) or detect an imbalance between supply and demand that suggests a future price curve slope. Historically, economic theories claimed the importance to take advantage of these information [4,10,14], but so far nobody to our knowledge had highlighted this fact experimentally.

Table 1. Chronological history of sent orders.

Id.	Sender	Direction	Price	Quantity
o1	Agent2	sell	111.5	8
o2	Agent1	sell	111.1	10
o3	Agent1	bid	110.6	6
o4	Agent3	sell	111.0	10
o5	Agent1	sell	110.9	8
o6	Agent2	bid	110.9	8
o7	Agent2	bid	111.0	7
o8	Agent4	bid	110.9	2
o9	Agent3	sell	110.9	2
o10	Agent4	bid	110.8	7
o11	Agent2	sell	110.8	5

Table 2. Chronological prices history, along with quantities exchanged and agents involved in the transaction.

Price	Quantity	Bid order	Sell order
110.9	8	o6	o5
111.0	7	o7	o4
110.9	2	o8	o9
110.8	5	o10	o11

Table 3. Order book state after matching orders from Table 1.

Direction	Order	Sender	Quantity	Price
Vente	o1	Ag2	8	111.5
	o2	Ag1	10	111.1
	o4	Ag3	3	111.0
bid-ask spread ↕				
Achat	o10	Ag4	2	110.8
	o3	Ag1	6	110.6

This paper main purpose is to demonstrate that it is possible to design trading behaviours that take advantage of all these information and offer a most efficient and effective trading behaviour.

1.2 Learning Trading Agents

As multi-agent systems, machine learning techniques have experienced a real boom in recent years. Main learning families (supervised, unsupervised, reinforcement) are based on different algorithms, the best known and most used are probably [12]:

- genetic algorithms,
- neural networks,
- Bayesian networks,
- support vector machines.

These techniques have been used in artificial markets with more or less success [2,11]. However, they all suffer the same explanatory default: once learning is achieved, it is difficult to understand why decisions are made, to highlight the cause of the outbreak of specific behaviors, and thus to avoid biases in learning contexts.

Different learning agent types have appeared within the literature in recent years. Even if these agents are designed to perform well on artificial markets, theirs learning is only focused on past prices to predict the possible future prices evolution.

As B. Arthur [1], we have chosen another learning technique that may be less common, but is much better adapted to the need of explanation: classifier systems or *Learning Classifier Systems* (LCS) [6]. These systems use a population of binary rules set by the designer that a reinforcement algorithm sort and that can possibly be modified by a genetic algorithm. Other techniques, such as those mentioned above would probably also work but our purpose here is not to make a comparison, but above all to show how to design an adaptive model and yet explanatory using the data set of one market to obtain varied and relevant behaviours.

One of the first artificial market, the SF-ASM (*Santa Fe Artificial Stock Market*, [1,15]), already used LCS for its reasoning agents, but this market is equational, and thus its agents could only take into account price history.

We show in this paper how to take advantage not only of past prices, but of all the available information in an order-driven multi-agent stock market simulator.

1.3 LCS Solely Based on Prices

Before discussing the overall complexity of a stock market, let's first describe the principle of a classical classifier system [1,9] by showing its usage in a macroscopic context where agents only study past prices to define their orders.

A LCS is initialized with a set of conditions on market state called indicators. These are the "sensors" used by the LCS to perceive market dynamic. Therewith, it is possible to derive a binary sequence whose length is equal the number of indicators used to characterize the current market state. Table 4 shows some indicators examples that can be used.

The first indicator `Ind1` is satisfied if the current price is higher than the previous price. `Ind2` is satisfied if the current price is higher than the average of the last 5 price, while `ind3` is satisfied when the current price is less than 100.

These indicators being fixed, each LCS has a set of rules (or classifier) consisting of a triplet *(condition, score, action)*

Table 4. Technical indicators based on past prices

Market indicator name	Market indicator definition
Ind1	$p_t > p_{t-1}$
Ind2	$p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$
Ind3	$p_t < 100$

Table 5. A LCS example using 5 rules

Rule	State S	Action A	Score F
R1	#10	bid	5
R2	1#0	sell	18
R3	00#	bid	12
R4	110	hold	4
R5	#11	sell	9

- the state S of a rule is a sequence of *trit* (***tr**inary dig**its***) that determines whether the rule can be activated given the current situation. These *trit* can be 0, 1 or #. In this sequence, each *trit* is an indicator. If a *trit* is 1 (resp. 0), the corresponding indicator must be *true* (resp. $false$) to enable the rule. The sign # means that the indicator is not be taken into account for rule activation.
- the ability to score F is the confidence we can have in this rule, based on its previous success prediction. The higher it is, the better is the rule.
- A action (bid or sell) to be performed if the rule is triggered. This choice is equivalent to a prediction on prices. Deciding to buy when we predict that prices will rise, and deciding to sell when we believe that prices will fall.

An LCS can have at most 3^n rules, where n is the number of indicators. Table 5 is an example of LCS with 5 rules using three indicators. For example, the first rule R1 can be activated if the current market situation is the state 010 or 110. This rule allows the LCS to select a purchase order, and his score is currently 5.

The LCS works as follows: every time he has to make a decision, it selects a rule among the activated rules with a probability proportional to the score of each rule. Then, the LCS performs the action associated with the selected rule (sending a signal to buy or sell). In the next activation of the LCS, the score of each activated rule will be corrected up or down depending on the accuracy of the prediction made.

In addition to the reinforcement system, it is common for a LCS to use also a genetic algorithm to renew its set of rules during simulation. It is applied regularly on the set of rules, using the score of each rule to produce natural selection within the classifier system. Rules whose score is below a threshold are eliminated, we cross and mute the best rules to regenerate the population.

The rules are not activated over time (due to conflicting indicators) are automatically eliminated. The combination of system update scores and genetic algorithm allows the LCS to achieve an effective learning.

There are other learning agents derived from LCS systems. For example, it is possible to make a social learning (information and learning are shared by multiple agents), or perform a hierarchical learning process multi-agents (HXCS [20]). However, LCS agents used here are agents of type XCS (*eXtended Classifier System*, [5,19]) realizing a simple reinforcement learning by updating the score of each rule.

2 An Order Based LCS

Adapting an LCS to an order-driven market poses several problems: temporal references within indicators may have different interpretations, in an order-driven market defining the direction (bid or sell) alone is not sufficient because it is necessary to also produce a proposal price and quantity, and finally, the agents must take advantage of information from pending orders contained in order books.

2.1 LCS and Temporal References

By adapting a classifier agent to a multi-agent system, the questions of the unity of time should be considered, because several indicators rely on this notion. Indeed, all agents can express themselves during a time step (in real time or turn to speak), but not always in the same order, and several prices may be fixed within one simulation step.

For example, if we want to know if the price has increased or decreased, the condition $p_t > p_{t-1}$ may not have the same meaning for each agent. In fact, agents have their own rhythms to place an order and a reference to past prices can be absolute or relative.

More precisely, to check the condition $p_t > p_{t-1}$, the agent can consider that t refers to either the known sequence of events by the market (the agent compares the current price at the last price fixed by the market), or the sequence of events known by the agent (the agent compares the current price at which prevailed the last time he made a decision, knowing that many prices have since been fixed).

We choose in this study to consider that the time is “the agent time”, that is to say the sequence of events known by the agent, because it allows agents to reason about values spaced in time according to their need.

2.2 LCS and Order-Driven Stock Markets

When an agent send an order to the market, the direction of the order is determined by enabled rule, but it is also necessary set a price limit and a quantity. For a fair comparison between agents, we propose to use the same policy of pricing and quantity for all agents.

For pricing strategies, we have decided to rely on the best prices contained in the order book. For example, for the order book presented within Table 3, the values are:

$P_{BestBid} = 110.8$ and $P_{BestAsk} = 111.0$

Using these best prices, we propose two different strategies:

- fixing the price to place the order at the top of the order book:
  $P_{Bid} = P_{BestBid} + \varepsilon$
  $P_{Ask} = P_{BestAsk} - \varepsilon$
- launch an order that will be immediately matched (at least in part), with a price equal to the best opposite order:
  $P_{Bid} = P_{BestBid}$
  $P_{Ask} = P_{BestAsk}$

Pour fixer la quantit, nous proposons deux stratgies: soit une quantit constante ($Q = k_c$), soit une quantit proportionnelle au score de la rgle active ($Q = k_p F$).

By combining these two types of strategy, it is possible to design four different policies, we compare them in Sect. 4. Thus, the agent LCS allows him to determine the direction while policies set a price and a quantity. An agent with these two elements is able to send orders to the market.

2.3 Leveraging Order Books State Information

To show that the learning agents can be improved by taking into account market microstructure, we propose to give to previous agents the ability to access order books information. To this aim, we add several indicators based on orders waiting within order books and show that these indicators give agents relevant information that improve their trading behaviour.

In macroscopic systems, the usual reasoning is to perform a technical analysis of historical prices to derive a future increase or decrease. This is typically the case strategies *chartist* that seek specific forms within price curves.

To show the contribution of order book information for agents reasoning, we propose in this paper to start with a reference LCS agent called `PriceLCS`, reasoning only on past prices, and improve it by adding agents called `OrderLCS` which rely on prices but also use market microstructure indicators. Then, we show through a set of experiments that these indicators provide relevant and useful information to agents to improve their trading performances.

The `PriceLCS` only uses technical analysis indicators presented in Table 6 that can take into account price change on both short and long term on three types of simple criteria: price evolution compared to previous price (indicator 1), the average compared to previous price n (indicators 2–4), or from the middle of the range of values of n previous prices (indicators 5 and 6).

We now want to compare `PriceLCS` with better informed agents, `OrderLCS` that benefit not only from prices but also from pending orders.

Indeed, order books contain a lot of information, in particular, the gap between the best bid and best ask, called *bid-ask spread* (see Table 3). This

Table 6. Technical analysis indicators shared by `PriceLCS` and `OrderLCS`

Id	Market indicator definition
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	$p_t > 1/2[Minp_i + Maxp_i]_{i\in[t-1,t-100]}$

Table 7. Technical market indicators based on bid-ask spread

Id	Market indicator definition
7	$bestP_{Ask} - bestP_{Bid} < k_7$
8	$r_t < k_8$

value can be related to an asymmetry of information, or a uncertainty about the value of the security. Moreover, it is a common market liquidity measure, the higher is the spread, the higher is the risk for an agent to sell or buy an asset in a short time frame.

To effectively leverage information from order books, we propose to add new indicators to `OrderLCS` agents: those concerning the value and evolution of the *bid-ask spread* and those related to an imbalance between supply and demand. The indicators proposed here are examples of criteria based on the orders that may be used by agents and are used in the evaluation Sect. 4.

Bid-ask spread. One can argue about the use of the *bid-ask spread* absolute value (indicator 7 in Table 7), but we propose to use instead the following ratio: $r = \frac{bestP_{Ask}}{bestP_{Bid}}$ (indicator 8).

One can for example compare the current value of this ratio to previous value (indicator 9), or the average of k_{10} previous values (indicator 10), or in the middle of the range of k_{11} previous values of r (indicator 11), to determine if the current *bid-ask spread* value is rather high or low (Table 8).

Table 8. Market indicators based on ration evolution $r = \frac{meilleurP_{Vente}}{meilleurP_{Achat}}$

Id	Market indicator definition
9	$r_t > r_{t-k_9}$
10	$r_t > 1/k_{10} \times \sum_{i=t-1}^{t-k_{10}} r_i$
11	$r_t > 1/2[Minr_i + Maxr_i]_{i\in[t-1,t-k_{11}]}$

Table 9. Market indicators based on relative size of bid and ask pending orders

Id	Market indicator definition
12	No bid orders pending
13	No ask orders pending
14	$\frac{nbAsk}{nbBid} > k_{14}$

Indicators Based on Imbalance Between Supply and Demand. The relative size in number of orders on both sides of the order book is also a useful information because it may reflect an imbalance between supply and demand (Table 9). This imbalance in a way or the other, may announce an upcoming change price that the agent could benefit. Indicators 12 and 13 check if there are buy and sell orders in the order book, and indicator 14 reasons on the ratio $\frac{nbAsk}{nbBid}$.

However, the relative size of the two parts of the order book is not the best way to assess the imbalance between supply and demand. In fact, if 10 assets are sold at the same price p_0, the offer is better than if only one asset is on sale for p_0 and 9 others have a higher price, yet $\frac{nbAsk}{nbBid}$ ratio remains the same. We should therefore also take into account the price differences in the calculation of supply and demand.

$$offer = \sum\nolimits_{order \in Ask} \frac{Quantity(order)}{Price(order) - bidAskMid}$$

$$demand = \sum\nolimits_{ordre \in Bid} \frac{Quantity(order)}{bidAskMid - Price(order)}$$

with:

$$bidAskMid = \frac{bestP_{Ask} + bestP_{Bid}}{2}$$

bidAskMid is the average of the best ask price and the best bid price. By dividing the quantity of each order by the difference between its price and *bidAskMid*, taking into account the fact that some orders have a price limit too high or too low to constitute an interesting offer or demand. We are then interested in the ratio $q = \frac{offer}{demand}$ (indicator 15) and its evolution (indicators 16–18).

`OrderLCS` agents use indicators 1–6 as `PriceLCS` agents, but they also use indicators based on orders (indicators 7–18). These order based indicators are only examples and many other indicators could be relevant. Nevertheless, the proposed indicators cover many aspects of order books and are widely accepted in finance. Indicators using a constant k have been implemented in several versions with several values of this constant.

Table 10. Market indicators based on imbalance within bid ask pending orders

Id	Market indicator definition
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	$q_t > 1/2[Minq_i + Maxq_i]_{i\in[t-1,t-k_{18}]}$

3 Methodology

Our study has two goals: to show that price based LCS are effective and that order based LCS are more effective. To do this, we first compare `PriceLCS` to various *chartist* agents to check that their learning mechanism allow them to get better results (first stage on the horizontal axis of the Fig. 1). In a second step, we want to demonstrate that indicators based on orders can improve LCS agents. For this, we compare `PriceLCS` with several `OrderLCS` agents (second stage on the horizontal axis in Fig. 1).

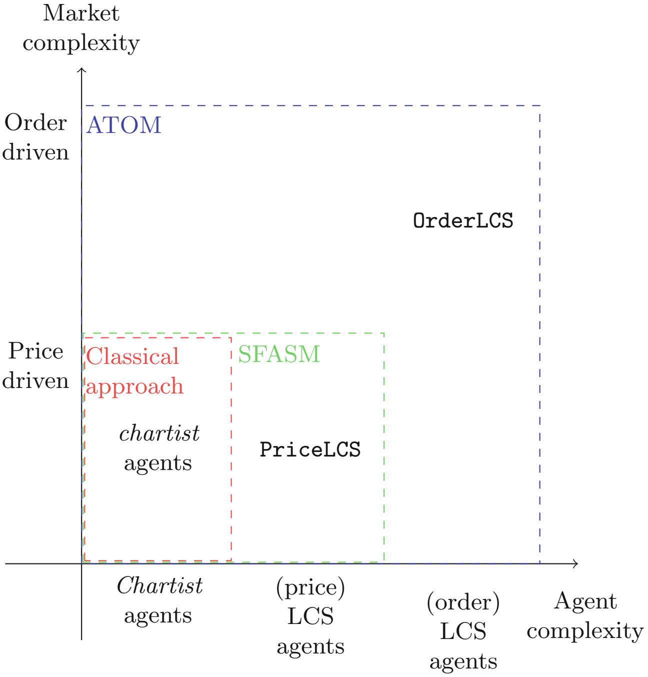

Fig. 1. Using a more complex market model allow the definition of better informed agents. We seek to demonstrate that LCS agents efficiency is better than chartists agents and then, that order based LCS agents outperforms price based LCS agents.

Evaluating and comparing agent behaviours is a difficult art. Firstly because, like voting systems, there are always methods that promote particular agents, and secondly because an agent is rarely good in absolute terms but often in relation to its opponents and the environment. Thus, we distinguish two problems: estimate an agent gain, that is to say its performance in a specific simulation, and to evaluate the agent, that is to say, estimating its performance being as general as possible.

3.1 Estimate Agents Performances

To compare agents, we must define a method to compute agents' gains, regardless of the type of evaluation selected. Two main criteria are possible here: either liquidity (*cash*) possessed by an agent or the richness of it (*wealth*), sum of its liquidity and the estimated portfolio value.

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

Although this estimation is questionable because it is approximative, it has the advantage of taking into account all the agent assets.

3.2 Evaluation Method

It is very difficult to say that an agent is better than another in absolute terms. In order to achieve the highest possible objective measures, it is important to test the agent in a sufficiently rich and representative selection to avoid potential biases due to a favorable or unfavorable environment. Moreover, the variation of the comparison set improves the objectivity of the measure. However, when an agent performs better than another in several sufficiently varied environments, then this agent demonstrated better robustness and we can assume that this agent is generally best.

Agent Comparison. Comparing agents by inserting them into a single environment is a major problem. It can exist specific relationship between two agents called "predator-prey" relationships, when these agents are in the presence of each other, the predator will maximize its results at the expense of prey. This, however, tells us nothing about the quality of each agent. If there is a relationship between two such agents that you want to evaluate, then the results for the two agents are distorted. This is why we use in our experiments a method that evaluates independently different agents to compare, based on the use of agents other than those to be evaluated. Finally, each type of agent should not be represented by a single individual, but by an agents population.

Populate Simulations. A good way to compare agents is to test their robustness to the variety of possible environments, and to average their performances. The performance obtained by an agent depends mainly on other competing agents, which is why it is necessary to vary types of agents encountered. To obtain diversified situations, simulations are populated with other agents types than those being evaluated. We call E all of these agents types, and S the set of agent types to evaluate. Some agents may be in S and E, if we try to evaluate them while they are also used to populate simulations. Agents that constitute E can be:

- *chartist* agents (moving average, RSI, momentum, variation, indicators and mixed moving average) using simple conditions on prices to predict price changes and decide whether to buy or to sell,
- periodic agents who buy and sell periodically
- *Zero Intelligence Traders* (ZIT, [13]) that send orders in a random direction with a random limit price,
- LCS agents based solely on price. These agents use simple market indicators (all of the form $p_t > p_{tX}$). These agents perform quite badly, but it seems important to us to have a minimum of adaptive agents within E.

Agent Families. To evaluate a trading behaviour, it is preferable to perform experiments with multiple instances of each agent instead of only one. In a real environment, it is rare that an individual is the only one to use a given strategy. So the concept of agent family, which is a set of agents using the same strategy and the same parameters has to be introduced. The use of agents families and the scaling that it induces has several consequences, including a better temporal distribution of the round of talk, possible interactions between multiple agents of the same type, or the smoothing of result for non deterministic agents.

3.3 Evaluation Methods

Once the measurement of individual performance is fixed, there are many ways to evaluate the performance of an agent in a community of n agents. Two main types of evaluation coexist [3]: evaluations in which n agents are placed in the same environment and compete with each other, and evaluations in which n agents are classified with respect to the same set of opponents. For each of these two types of evaluation, it is still possible to evaluate agents individually or with agents families (agents with the same type and parameters).

Various performance measures are then possible:

- n agents are evaluated in the same simulation. Agents are classified according to their exact gains at the end of the simulation.
- n agents families are evaluated in the same simulation. Agents are ranked on the average earnings of their families at the end of the simulation. Using agents families, results variability due to agents stochasticity is attenuated.

- each agent family is evaluated separately from the other by running it against a common reference population. The n agents are then classified on the average earnings of their families. Separating families that are compared avoids introducing a bias in the evaluation due to the dominance of one agent over another.
- agent families are evaluated against a common reference population within an ecological competition. At the beginning, all families have the same number of instances but at the end of the simulation, instances are adjusted according to their family score.

These different simulations are ranked on their pertinence, indeed each protocol is superseded by its followers. For example, the use of reference families avoids distorting the comparison between families who would have dominance relationships between them even if one is not better than the other. In this article, we have chosen the most complete of them, the ecological competition, because it introduces a variation of population that avoids mutual support between families of agents and thus ensures a better robustness of the results.

3.4 Ecological Competition

Ecological competition is a selection method inspired by biology and the natural selection phenomenon [16,18]. In such context, several families evolve in the same environment, such as animal or plant species sharing the same environment. As in nature, their populations vary over time, in such a way that the best families see their populations increase, while the others decline.

In our case, a competition is a series of simulations on the market. Each family begins with an identical number of individuals, the total population of the competition remains constant throughout the competition. After each generation, the population of each family is revalued based on its gain. After each generation a proportionality rule on the score of each family is applied to keep the total population constant.

A family score is the total gain of its instances during the simulation. But, in ATOM, as in real markets, *traders* have the opportunity to borrow money to purchase assets. The agent *cash* can then be negative as well as its *wealth.* It is therefore possible that a family has a negative total gain, which is a bit problematic to apply a proportionality rule. To solve this problem, we propose to subtract the gain of the worst agent p (gain of p are most often negative, then it will be an addition) to the gain of all agents a.

$$gain_a \geq gain_p \Rightarrow gain_a - gain_p \geq 0$$

In this way, the gain of p agent is reduced to 0 and that of all others is positive (as the total gain of each family). The total gain of a family f is the sum of the modified wealth of its agents. It is then possible to apply a proportionality rule for calculating populations of the various agent families.

The population volume of a given agent family at the end of an ecological competition represents its adaptation to a particular environment (other families competing agents) and its effectiveness in this environment.

It may seem surprising at first that ecological competition is preferred to a more classic economic competition, where an agent leaves the competition when it is ruined. However, this approach presents several problems:

- with the ability to borrow, it is difficult to determine whether an agent is ruined, this choice is arbitrary.
- an agent can decide to leave the market before going broke on it.
- this approach does not allow the opportunity for new agents to enter the market by playing a strategy they observed success.

Ecological competition has the advantage of presenting a dynamic environment that adapts itself to agents performances. This allows to experience a robustness against evolving environments.

3.5 Experimental Protocol

When an agent performs better than another in several various ecological competitions, we can assume that this agent is generally best. This is the method we used to evaluate our agents.

For each agent type to be evaluated $s \in S$, the following procedure is performed. 500 ecological competitions populated with one agent family of type s and various families of agents (5–20) from E (which contains currently forty agents), randomly selected each competition. 50 generations competition are generally sufficient to reach stable populations volumes in most cases. Each generation is composed of a trading day made of 2000 decisions per agent, which is necessary for learning agents to adapt themselves to their environment.

The average population of each family obtained after 500 competitions is a good measure of the performance of an agent, because it takes into account the robustness of the variety of possible environments, and the robustness to adaptive environments.

The first experiments consists in assessing the effectiveness of price bases LCS agents before comparing them to order based LCS agents.

3.6 Price and Quantity Policies Evaluation

The first experiment is to determine which pricing and quantity policy should be used with our LCS agents. To this aim, we compare a price based LCS with all policies variation against a reference population. As shown in Fig. 2, this experiment shows that the policy giving the best results is the policy that sends orders placed at the top of the order book, with a constant quantity. Thus, these policies are used throughout all other experiments.

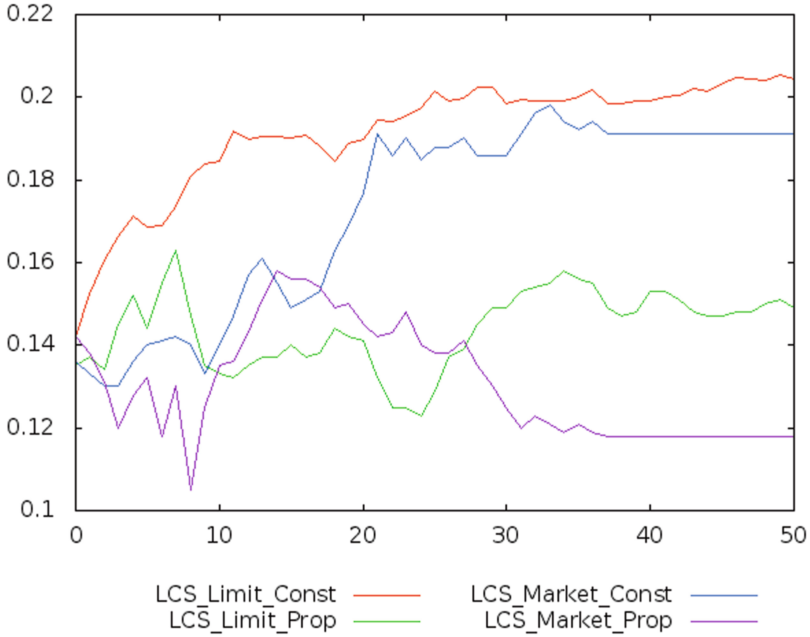

Fig. 2. Average PriceLCS family proportion with respect to the price and quantity fixing mechanisms: *Limit* to be at the top of the order book, *Market* for an immdiate execution, *Const* for a constant quantity and *Prop* for a proportional quantity.

4 Simulations Results

4.1 Price Based LCS Agents

The second experiment assesses the quality of price based learning. Figure 3 represents the average proportion of the population of various agents in an ecological competition. One of these families is `PriceLCS` and other agents are *simple* chartists (moving average, RSI, momentum ...) or agents with basic behaviours (periodic ZIT). It should be noted that price based LCS agents outperforms other families. Thus, this experiments demonstrates the effectiveness of the learning process achieved on prices analysis.

4.2 Order Based LCS Agents

The third experiment check that order based LCS agents perform better than LCS agents based solely on prices. Figure 4 represents the average population of a few agent families in an ecological competition. The `PriceLCS` uses indicators based on prices described in Table 6. For others families, these same price based indicators are used, but some order based indicators described in Tables 9 and 10 are added. 298 different agent types based on these additional descriptors have been tested, the Fig. 4 gathers only those that have obtain the best results.

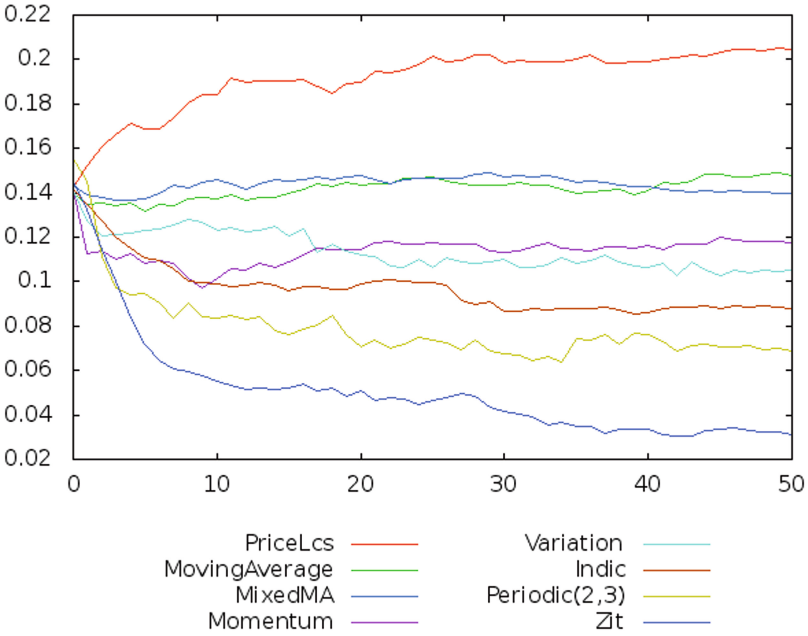

Fig. 3. Average population volume for each family: the `PriceLCS` graph is for price based LCS agents

It was found that the best families use 1–3 indicators taken from instances of indicators 10, 11, 16 and 17 (see Table 11). Other indicators have little influence or none at all on results. Indicators used by each family are described in Table 11. Each bit corresponds to a line. For example, agents of the family `OrderLcs_10001` use indicator 10 with $k = 100$ (first line table) and the indicator 17 with $k = 100$ (fifth row of the table).

We also observe that many of these families do better than the base agent. Agents using these good indicators have an average population at the end of the competition of about 100 % that of the base agent. This experiments demonstrates that LCS agents can be improved by the addition of order based indicators.

4.3 Assessing an Indicator Utility

We propose an utility measure u_i of an indicator i as the average score of rules for which the *trit* question is not undetermined (i.e. #). This score is calculated on all the rules of agents of a family at the end of a simulation.

If more indicators are used, the average utility of an indicator is small. To compare the usefulness of two indicators, we use the following ratio:

$$s_{ind} = \frac{u_{ind} \times nbIndicateurs}{\sum\limits_{i \in indicateurs} u_i}$$

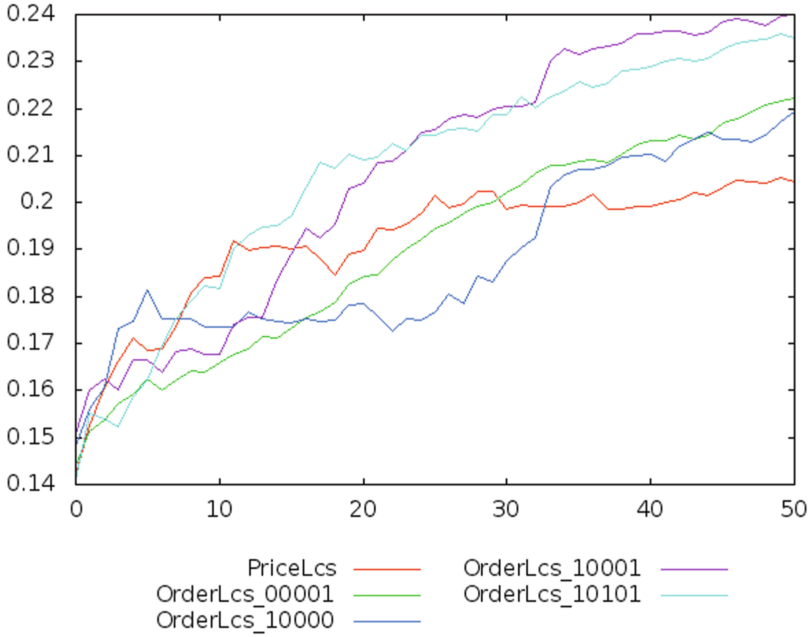

Fig. 4. Average volume population in an ecological competition: the `OrderLCS` graph is for order based LCS agents.

Table 11. Order based indicators that improve `PriceLCS` agents

Id.	k	Indicator instantiation
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	$q_t > 1/100 \times \sum_{i=t-1}^{t-100} q_i$

If $s_{ind} > 1$, then u_{ind} is higher than the average useful indicators of the observed LCS. Otherwise, if s_{ind} is less than 1, it means that u_{ind} is below the average. Therefore an indicator of quality maximizes the ratio s. We note on the Table 12 that this ratio is less than 1 for an indicator randomly returning true or false (line 1), less than 1 for indicators based solely on prices (line 2), but greater than 1 for several indicators based on orders (line 3), in particular those listed Table 11.

Table 12. Ratio value s for several kind of indicators

	Market indicator definition	s
1	Random	0.860
2	$p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$	0.961
3	$r_t > 1/100 \times \sum_{i=t-1}^{t-100} r_i$	1.034

4.4 Learning Mechanism Saturation

The more information an agent has, the greater is its potential to achieve good prediction on price evolution. Nevertheless, the addition of an indicator expands the search space, and makes the learning process slower and sometimes less effective. All information are not relevant or pertinent. For example, one can think that $p_t > p_{t-1}$ is more relevant than $p_t - 100 > 101 - p_t$ for many situations.

Indeed, the expansion of the search space is the same regardless of the indicator added, so the information provided by an indicator must be significant enough to compensate for this expansion. In addition, if two indicators provide similar information, the added value of the second is low. We found that our LCS agents must therefore have a limited number of indicators (between 6 and 9), sufficiently differentiated from each other.

5 Conclusion

To achieve realistic financial simulations, it is important to populate artificial markets with adaptive agents. It allows to obtain price curves or price histories that are more realistic and consistent with stylized facts found in computational finance. It also enhances the experience of human agent who participates in a simulation with agents more *intelligent* and robust. It finally allows you to test the rules of market regulation in richer and more lifelike environments.

However, until now, the lack of software platforms both running orders driven markets and implementing a multi-agent approach, this type of simulation was done only on the price curve generated through equational models, as done by B. Arthur with SF-ASM. The ATOM platform, by its fidelity to order driven markets like *EuroNEXT-NYSE*, helps to push learning agents much further.

Learning in the context of a agent based artificial stock market is a complex process. It is often believed to one can learn from the behaviour of other agents. Recognizing other agent behaviour is a difficult thing, trying to adapt itself to an evolving environment is even more complex. It is also often believed that there is nothing to learn from stochastic agents and that only agents with deterministic behavior can bring more information. These suppositions do not take into account interesting information that are brought by the microstructure of double order books who sets prices in order driven markets. They plays a role similar to an accumulator. These are not necessarily the last orders to arrive that will be executed first, but the best deals. It is then possible make effective use of the information contained in these order books and have an *edge* on price trends. From these information can be found the best bid or best offer for sale of course, but also the size of the *bid-ask spread* or available quantity available, weighted or not by its distance.

In this article, after detailing the various possibilities to reason about orders and their consequences, we have shown how to set up learning classifier systems that take into account the information present in the market microstructure, that is to say pending orders in order books. To compare these agents, we have implemented an original adaptation of the principle of ecological competition that allows us to measure an agent performance and also its robustness to environmental changing. We were able to show that an agent who studies pending orders in order books is far more efficient than agents *chartist* or as his counterpart that reason solely on prices.

Further work is certainly needed in this direction, by varying the learning methods used and also by introducing indicators that attempt to recognize typical trading behaviours by studying orders evolution from a specific agents within order books. This, we consider this work as a first step towards more complete adaptive learning behaviours for artificial agents.

References

1. Arthur, W.B., Holland, J., LeBaron, B., Palmer, R., Tayler, P.: The Economy as an Evolving Complex System II, pp. 15–44. Addison-Wesley, Reading (1997)
2. Barbosa, R.P., Belo, O.: An agent task force for stock trading. In: Demazeau, Y., Pěchouček, 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)
3. Beaufils, B., Mathieu, P.: Cheating is not playing: methodological issues of computational game theory. In: ECAI'06 (2006)
4. Belter, K.: Supply and information content of order book depth: the case of displayed and hidden depth (2007)
5. Boland, E., Klingebiel, K., Stodgell, T.: The xcs classifier system in a financial market (2005)
6. Booker, L.B., Golbergand, D.E., Holland, J.I.: Classifier systems and genetic algorithms. Artif. Intell. **40**, 235–282 (1989)
7. Brandouy, O., Mathieu, P.: Efficient monitoring of financial orders with agent-based technologies. In: Demazeau, Y., Pěchouček, 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)
8. 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)
9. Brenner, T.: Chapter 18 Agent Learning Representation: Advice on Modelling Economic Learning. Handbook of Computational Economics, vol. 2, pp. 895–947. Elsevier, Amsterdam (2006)
10. Cao, C., Hansch, O., Wang, X.: The informational content of an open limit order book. In: EFA 2004 Maastricht Meetings

11. Cao, L., Tay, F.: Application of support vector machines in financial time series forecasting. Omega: Int. J. Manage. Sci. **29**, 309–317 (2001)
12. Cornuéjols, A., Miclet, L., Kodratoff, Y.: Apprentissage Artificiel Concepts et Algorithmes. Eyrolles, Paris (2002)
13. Gode, D.K., Sunder, S.: Allocative efficiency of markets with zero-intelligence traders: market as a partial substitute for individual rationality. J. Polit. Econ. **101**, 119–137 (1993)
14. Kozhan, R., Salmon, M.: The information content of a limit order book: the case of an fx market (2010)
15. LeBaron, B.: Building the santa fe artificial stock market. Brandeis University (2002)
16. Lotka, A.J.: Elements of Physical Biology. Williams and Wilkins, Baltimore (1925)
17. Mathieu, P., Secq, Y.: Environment updating and agent scheduling policies in agent-based simulators. In: ICAART'2012 (2012)
18. Volterra, V.: Variations and fluctuations of the number of individuals in animal species living together. In: Chapman, R.N. (ed.) Animal Ecology. McGraw-Hill, New York (1926)
19. Wilson, S.: Classifier fitness based on accuracy. Evol. Comput. **3**, 149–175 (1995)
20. Gershoff, M., Schulenburg, S.: Collective behavior based hierarchical XCS. In: Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '07, London, UK, pp. 2695–2700. ACM, New York (2007)