Testing double auction as a component within a generic market model architecture

Julien Derveeuw and Bruno Beaufils and Philippe Mathieu and Olivier Brandouy

September 2007

1

Testing double auction as a component within a generic market model architecture

Julien Derveeuw, Bruno Beaufils, Philippe Mathieu, and Olivier Brandouy

Laboratoire d'Informatique Fondamentale de Lille
USTL - Cite Scientifique, 59655 Villeneuve d'Ascq, France
{derveeuw,beaufils,mathieu}@lifl.fr

Lille Economie et Management
USTL - 104 Avenue du Peuple Belge, 59043 Lille, France
olivier.brandouy@univ-lille1.fr

Summary. Since the first multi-agents based market simulations in the nineties, many different artificial stock market models have been developped. There are mainly used to reproduce and understand real markets statistical properties such as fat tails, volatility clustering and positive auto-correlation of absolute returns. Though they share common goals, these market models are most of the time different one from another: some are based on equations, others on complex microstructures, some are synchronous, others are asynchronous. It is hence hard to understand which characteristic of the market model used is at the origin of observed statistical properties. To investigate this question, we propose a generic model of artificial markets architecture which allows to freely compose modules coming from existing market models. To illustrate this formalism, we implement these components to propose a model of an asynchronous double auction based on an order-book and show that many stylized facts of real stock markets are reproduced with our model.

1.1 Introduction

Artificial stock markets are models designed to capture essential properties of real stock markets in order to reproduce, analyze or understand market dynamics with computational experiments. Despite research advances in modern finance many questions remain unsolved: market dynamics exhibit, for instance, particular statistical properties, called *stylized facts*, which origins are not clear. As real markets are complex systems, it is really hard to study them directly because too many parameters stay out of control. Hence, multi-agents simulations of these markets seem to be a key for a better understanding of their properties.

Building such models implies to simplify reality as most as it can be in order to keep markets most representative and characteristic features. In the litterature (see

[0] This work is supported by european funds of FEDER and the CPER TAC of Nord-Pas-de-Calais region.

for example (Baron, Arthur, and Palmer 1999), (Cincotti, Ponta, and Pastore 2006) or (Ghoulmie, Cont, and Nadal 2005)) real markets structure complexity is often circumvented by the use of an equation weighting the balance between bids and offers as a price formation model. This simplification is in complete contradiction with the reality of stock markets where prices *emerge* from agents interactions through an order book which do not act as a central weighting entity but as a peer-to-peer meeting point used by agents to exchange stocks. However, such studies manage to reproduce realistic price series, which seems odd regarding market models used. We can then wonder if some of these models are more suited than others to capture market dynamics.

To answer this question, it seems that a comparison between these models needs to be realized in order to put them to the proof and investigate their robustness. Hence, we propose in this article a generic market model architecture based on four independant entities, each of which can be modelized in different ways. We show that existing models found in litterature fit well in this architecture. We then propose an artificial stock market model which takes into account real markets characteristics: trading activity takes place *continuously* through an *asynchronous* mechanism. Agents interact through the market by posting *orders* in an *orderbook*, as it happens on real market places. We show that without making any strong assumption on agents behaviors, this model exhibits many statistical properties of real stock markets.

1.2 Quick review of different ASMs architectures

Since the first artificial stock market was developped in the early nineties at the Santa-Fe Institute (Palmer, Arthur, Holland, LeBaron, and Tayler 1994), many market models have been developped. Though almost all of them aim to reproduce the same market properties (the so-called stylized facts) with the same multi-agents simulation methods, they all exhibit different properties: some are synchronous, while others are asynchronous. Some of them require agents to emit realistic orders (direction/price/quantity) while others only require a direction (buy/sell) to compute the new stock price. Without pretending to be completely exhaustive, we investigate in this section some of these models in order to identify the most represented microstructures and trading rules in artificial stock markets.

The Santa-Fe artificial stock market

Historically, the first model to be developped was the *Santa-Fe Artificial Stock Market*. This model is mainly characterized by the use of a macroscopic equation based on demand and supply law to compute the new traded stock price. Hence, agents take their decisions synchronously and emit their desires as a direction $a_{i,t}$ (buy $a_{i,t} = 1$ or sell $a_{i,t} = -1$) to the market, which calculates the imbalance between demand and supply ($I_t = \sum_i a_{i,t}$), to finally compute the price according to equation 1.1.

$$p_{t+1} = p_t(1 + \beta \times I_t) \tag{1.1}$$

Though this model may seem attractive due to its relative simplicity, its lack of realism regarding real market microstructure is obvious: agents take their decisions synchronously without being able to reason about others beliefs; moreover, agents are not even aware of the quantity of stocks they will trade due to the clearing process used to realize exchanges between agents once the price is calculated.

The $-game

To solve the question of market clearing, a possible solution is to add a market maker to the model, so agents are always satisfied with the quantity they want to trade. This feature was incorporated in the $-game ASM (Andersen and Sornette 2003). As the market maker provides liquidity to the market (e.g. he buys excess stocks and provides supplementary stocks when needed), his position needs to be covered to avoid bankruptcy. Hence, Andersen et al. use in their model a slightly modified version of the previous price calculation equation. Instead of only considering the current imbalance between demand and supply, they also take into account the global imbalance since the beginning of the simulation, which is the market maker current position. Using the same naming as above, the price update equation is then given by 1.2.

$$(\ln(p_t) - \ln(p_{t-1})) = \frac{I_t + \sum_{i=0}^{i=t-1} I_i}{\lambda} \tag{1.2}$$

Though this model correctly addresses the problem of stock liquidity and market clearing, it can't be considered as a realistic one: agents still interact synchronously with the market and only emit a desired quantity to trade, without having the ability to associate it with a desired price for the transaction.

The Genoa artificial stock market

To bring more realism to synchronous models, researchers from Genoa proposed a model called the *Genoa artificial stock market* in which agents are allowed to emit classical limit orders to the market (see (Raberto, Cincotti, Focardi, and Marchesi 2001), (Cincotti, Focardi, Marchesi, and Raberto 2003) or (Raberto, Cincotti, Focardi, and Marchesi 2003)). In this model, agents still take their decisions synchronously, but as they associate a limit price to the desire they pass to the market, a different clearing mecanism needs to be used to ensure that agents do not buy or sell stocks for a different price than the limit they asked for. This is achieved by computing a clearing price, which is defined as the crossing of the demand quantity curve function of price and of the supply quantity curve function of price (see equations 1.4 and 1.3 for a definition of these two series).

$$f_{t+1}(p) = \sum_{u|p_u \geq p} q_u^b \tag{1.3}$$

$$g_{t+1}(p) = \sum_{v|p_v \geq p} q_v^s \tag{1.4}$$

Though this model is more realistic than the previous ones, it still lacks an essential feature of real markets microstructure: the asynchronism of transactions.

Toy model of an asynchronous double auction

In order to get a more realistic time handling process in artificial markets, some researchers proposed models in which transactions take place asynchronously. This is the case of the toy model proposed in (Bak, Paczuski, and Shubik 1996). In this model, there are only $\frac{N}{2}$ stocks on the market, where N is the number of agents. Agents do not have the right to own more than one share at a time. They can therefore be sellers if they own a share, or buyers if they own nothing.

At each time step, an agent is given speak randomly and has the possibility to emit a desire according to the pre-cited rules. This desire is a composed of a price and a direction. If this agent finds an other one who is willing to make the opposite transaction with a compatible price, they immediatly exchange one share. If no counterparts are available, the agent's order is saved in a list until a counterpart is found.

Even if it is a toy model, this model is one of the first to take into account the asynchronism of exchanges on real market places. Agents act in a random order and a simplified orderbook is used to save agents desires. A criticism which can be made is that the market rules used (an agent can at most own one share) tend to make the market illiquid and prevent from testing realistic investment strategies.

We have seen in this section that many different market models are used to reproduce high frequency dynamics from real stock markets. Despite of their heterogeneity, they are used to reproduce the same three main stylized facts: the shape of the return distribution (which is fat-tailed and leptokurtic), the autocorrelation of absolute returns and clustered volatility. We can notice strong differences in the way agents express their desires, in the set of informations they are able to get from the market, and in the way they are given speak by the market. This a major problem regarding our main goal, which is to be able to compare heterogeneous market models in similar experimental environment.

1.3 A generic market model architecture

In the previous section, we have presented some of the most representative market models microstructures and trading rules found in the litterature. Their diversity is so great that it seems difficult to correctly identify which of these models parts are responsible for the statistical properties of computed price dynamics: are they due to the microstructure of the market ? to the way time is handled ? to the agent investment strategies ? In order to address these open questions, we expose in this part a generic model of market architecture which allows to unify these different models. We also show that this formalization allowed us to develop a concrete implementation of this generic architecture, which will make us able to compare artificial stock markets.

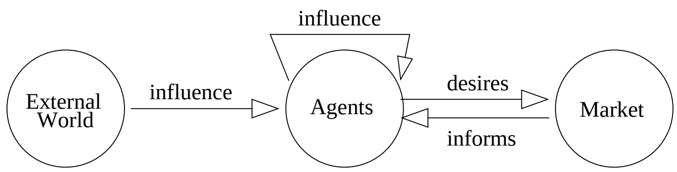

Fig. 1.1. General market model architecture

1.3.1 The abstract generic model

If we look at how markets operate, we can decompose them in three parts: the *market*, which allows agents to exchange stocks, *agents*, who trade through this market, and the *external world*, which can for example influence agents with informations. This situation is summed up in figure 1.1: agents communicate their desires to the market, being influenced by their peers or exogenous informations. They can also be influenced by public informations available from the market. If we make a parallel between this abstract model and multi-agents models of markets, we can see from the previous section that each of these three components can be modelized in different ways: the market can be an averaging equation or a complex microstructure; agents can be either cognitive, reactive or replaced by equations.

1.3.2 The concrete generic model

In order to experiment the influence of each of these modules on price dynamics, we need to be able to compose heterogeneous modules coming from the litterature. For example, to investigate the influence of market microstructure on prices, it seems interesting to study some of their different implementations for a given set of agents behaviors. Unfortunately, as we have seen in section 1.2, most of market models require agents to emit their desires in many different ways: there are sometimes expressed as a direction, sometimes as a quantity or even as limit orders. Hence, it seems obvious that to make our generic architecture practical, we need to propose some concrete details on its implementation.

Informations

In our formalism (see figure 1.1), we showed that agents were able to use some informations coming from the market in order to take a decision. As we saw in the first section, informations required by agents or published by market models are heterogeneous: some market models only publish the last transaction price, while others make all of the agents current positions public. Hence, to be able to compose any agents model with any market model, it is necessary to define the maximum set of informations needed by agents models and to define how all of these informations can be approximated when they are not present in a given market model.

According to our litterature review, agents use *at most* the following informations from the market:

- the last transaction price, which is an information available on every market model
- other agents desires (which represent the orderbook in asynchronous models).
- current demand and supply desequilibrium, which is available in most synchronous models. In asynchronous model, it is easy to deduce this information from the current orderbook state by summing quantities available in both sides of the orderbook.

To be able to compose any market model with any agent model, we have to define a set of translators able to fill missing informations from some market models if it is required by the agents. An example of such a translator (or wrapper) is described in table 1.1. Though we provide in our framework a full set of information translators which allow to translate any type of emitted information in any type of required information, the effect of these translators on experimental results still has to be investigated.

emitted by market	required by agent	translator description
(price, agents positions)	(price, desequilibrium)	translator sums up quantity associated to agents positions in order to compute the global demand/supply imbalance

Table 1.1. An example of information translator

Agents desires

In figure 1.1, we identified that agents emit trading desires to the market, which are then interpreted according to market model trading rules. These desires, in artificial markets as well as on real ones, are defined by a composition of the three following characteristics: a direction, a price and a quantity. Obviously, the direction is the minimal requirement in order to get a valid desire (emitting a desire to a market without saying if one wants to buy or sell makes no sense). The two others desires properties (price and quantity) are optional according to the agent or market model. As we would like to compose any agents and markets models which emit or require different types of desires, we need to define a translation system to make this composition possible.

Assuming that a direction is the minimum required to express an economic desire and that the maximum is a direction, a price and a quantity (which was deduced from our intensive litterature investigation), it is possible to propose a first set of translators (which are called wrappers in computer science) that are required to allow communication between any agent model and any market model. The effect of these wrappers on agents and market behaviors still has to be investigated. An example of such a translator is described in table 1.2.

emitted by agent	required by market	translator description
(d, , q)	(d, p, q)	interpret order as a market order, and fill the missing price with the best offer in opposite direction

Table 1.2. An example of desire wrapper

Time handling

In addition to the differences between informations required or emitted by the different modules of a market, time handling is managed in very different ways regarding the market model used: some are synchronous while others are asynchronous. Moreover, each of this time handling philosophy can be implemented in several different ways. These differences are a major problem to solve while trying to compose heterogeneous modules: if an agent strategy is built to operate in an asynchronous context, is it possible to make its strategy make sense in a synchronous one ?

To address this problem, we have splitted time handling from the market module and separated it in what we call a *simulation engine*. This additional module is reponsible for giving the ability to talk to the agents and for making the market treat agent desires when it is time to do so. For example, a synchronous simulation engine will give to all of the agents the ability to talk, and will then ask the market to compute the new stock price, whereas an asynchronous one will perhaps pick randomly an agent and then immediatly ask the market to take his desire in account.

Global framework layout

Due to lack of space, we can not explain further all of the implementation details that are needed to allow free market modules composition. Figure 1.2 sums up the general layout of our simulation framework, which we detail step by step:

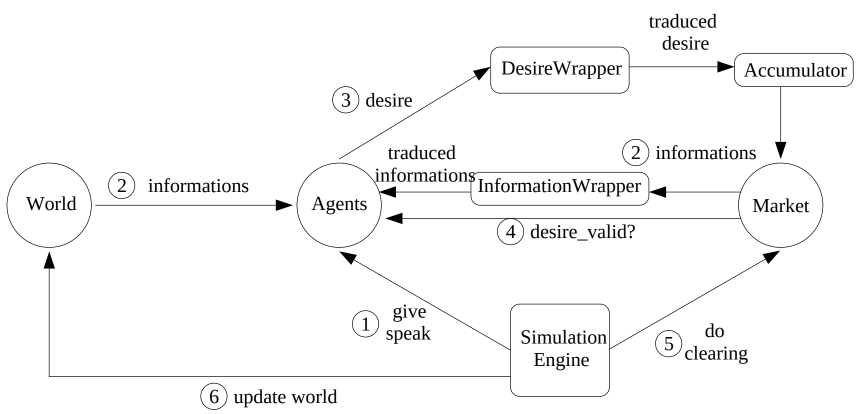

Fig. 1.2. Framework functioning

- *step 1*: The simulation engine gives speak to the agent(s) who are allowed to speak at current time according to the time policy in use.
- *step 2*: Before taking a decision, agents are able to ask the market some informations about its current state (best offers, current stock price, demand/supply imbalance, etc). As each market model can exhibit different public informations, they need to be treated by a wrapper which traduce them so they can be used by any agent model. Agents can also ask external world about its current state if their decision making process requires such an exogeneous information.
- *step 3*: Once agents have sufficient informations to take their decisions, they can emit a desire to the market. As we have seen before, this desire can be expressed in many different ways, so it needs to be traduced by a wrapper to be understood by any market model. These desires are then stored in an accumulator, which is useful to keep track of agents desires, in particular if the simulation engine is synchronous.
- *step 4*: Each time the market receives an agent desire, it immediatly informs the emiter about its validity. This is required as some market models require agents to meet specific conditions to be able to emit desires.
- *step 5*: Once the simulation engine has given speak to the agents allowed to do so, it notifies the market that it is time to take the agents desires into account. If the market is orderbook based, this means “insert new desires in the book”, whereas in equation-based models, this means “enter in a clearing phase and compute a new price”.
- *step 6*: The simulation engine finally gives the possibility to the world model to update itself.

Limitations

Even if our generic architecture is implemented and practical, it still has some limitations inherent to the major differences between models we try to compose one with another.

For example, some information translators need to be able to translate an information expressed as a single price in an information expressed as other agents positions. Even if other agents positions may be assimilated to the current stock price, impact of such translations on agent trading strategy have still to be investigated. The same observation can be made about the composition of agents designed to work in an asynchronous context with market models designed to work in a synchronous one.

Hence, our generic architecture still has to be improved and validated with intensive experiments, in order to make sure that translators do not bias simulations results. Even at this early stage, this generic architecture can however be merely considered as a formalism able to describe any artificial stock market model through their components.

1.3.3 An example of application: the market component as double auction

We have seen in section 1.2 that most of existing market models lack realism: some do not respect real markets asynchronism while others over simplify the way agents emit desires to the market. In consequence, we choose to illustrate the use of our generic market simulation framework by implementing a simple asynchronous double auction model following our formalism. This model can be linked up to the one used in (Raberto, Cincotti, Dose, Focardi, and Marchesi 2005). We will detail in this section how each module is defined according to the formalism we presented before.

The market component

The market component is a classical orderbook similar to the one used on market places such as EURONEXT. This orderbook requires agents desires to be expressed as a *direction*, a *price* and a *quantity*, which defines an order. These orders are all *limit prices orders*, which means that the price associated to the order is the maximum (respectively minimum) price the agent is willing to buy (sell) stocks. When an order is received by the market, it is stored in the orderbook according to price and time priorities if it has no counter part. When a counterpart is found, a transaction occurs immediatly and the price of this transaction is published.

The simulation engine component

In orderbook-based markets, time handling does not follow the same logic as in equation-based ones: central quotation system does not aggregate agents decisions at particular time steps and market participants are free to talk when they want. Hence, we need to implement the simulation engine component as a process which asks agents to speak asynchronously and which asks the market to update its current state each time an agent has spoken.

Our choice is to give randomly an agent the opportunity to talk regarless to the fact he has already spoken or not. The major inconvenient of this method is that some agents can be out of the market (have never the opportunity to speak) because of the random generator used in the scheduler. However, on real markets, some agents are very active (speak a lot) whereas others rarely interact with the market. For these reasons, this is the scheduling principle we choose.

The agent component

Following the works of (Gode and Sunder 1993), our agents are designed as purely reactive ones (as simple as possible), which implies that we do not make any strong hypothesis about the agents reasoning capabilities, nor on the information set they use to take their decisions, as it is done in most of other studies. The choice of using simple agents behaviors in this article is hence deliberate: our goal, here, is not to design realistic agents but to validate our microstructure model separately from the two other components of the market architecture.

These agents can be assimilated to *zero intelligence traders* who post orders with a random direction, a random price for a random quantity of stocks. When an agent emits a new order, he stops emitting new ones until his order is fulfilled or until the order reached his *timeout*. This *timeout* is randomly assigned to each agent at the beginning of the simulation and stays constant over time. This mainly guarantees that an order with a price too far from the current limits of the orderbook won't stay in it for an endless time.

1.4 Experiments

We present in this section some experiments we have designed to test our generic framework. Only a part of the statistical tests we made are reproduced here due to lack of space. Full experimental tools and results used to produced data presented in this paper may be downloaded at `http://cisco.univ-lille1.fr/papers/ae2007`. This experiments are realized using the market model, agents and scheduler exposed in previous section. All of our experiments are run on 20 000 time steps with 100 agents.

First, we interested ourselves to the returns distribution as its shape is one of the major characteristic of real price dynamics. This distribution, on real markets, is leptokurtic and exhibits fat tails. Table 1.3 shows some statistical results: the excess kurtosis measured oscillates around 4.5 which is similar to what can be observed with real markets data (see right column for a comparison). To further illustrate this property, figure 1.4 shows one of our experimental returns distribution compared to a theoretical normal distribution.

Description	Value (experimental)	Value (real data)
Excess kurtosis	4.52	4.158
Aug. Dickey-Fuller	-20.47	-18.47
ARCH	100%	100%

Fig. 1.3. Statistical results obtained with our interaction-based model, compared to the one obtained on real data (BMW daily stock returns coming from DAX30)

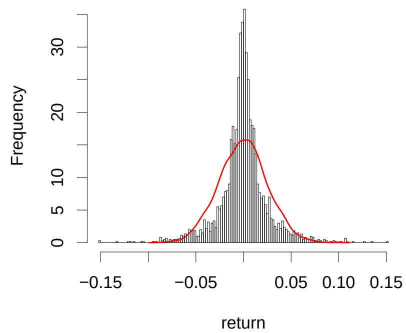

Fig. 1.4. Experimental returns distribution compared to a theoretical normal distribution with same mean and variance

Another major characteristic of returns is that they do not exhibit significant autocorrelation but that a short-range autocorrelation decaying over time exists when looking at their absolute value. Figure 1.5 presents the ACF plot for both returns and absolute returns. Comparing them to the ones obtained with real market data, we can see that returns properties similar to reality can be obtained with our interaction-based model. These properties are further verified by the use of the Augmented Dickey-Fuller test which tests for the null hypothesis "*The serie has a unit root*". Table 1.3 shows its result on our time series : the presence of a unit-root is rejected at a high confidence level as with real data (right column).

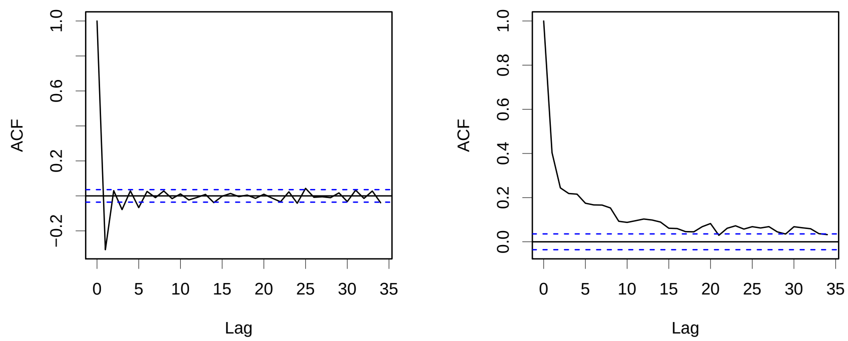

Fig. 1.5. ACF of returns and squared returns obtained in our experiments

We have seen in this section that time series obtained with our model exhibit the same statistical properties as real data sets. This results improve the preliminary ones obtained by (Raberto, Cincotti, Dose, Focardi, and Marchesi 2005). This shows that our asynchronous and continuous auction model is able to reproduce most of markets characteristics without making any asumption on agents behaviors or on an external world model.

1.5 Conclusion

In this article, we introduced a generic architecture of artificial market models. This architecture is composed of four independant parts: a model for the external world, another for agents behaviors, one for the market structure and a last for time handling. We have shown that most of existing market models can fit in this architecture, so it can therefore be considered as a description formalism of artificial stock markets. Moreover, our generic architecture allows to compose existing market and agent models, which is a major benefit if one plans to compare market models between them: it is now possible to do such comparisons in identical environments (e.g. with the same agents) and to draw strong conclusions from these experiments, which was not the case before. However, some of the effects of our generic model still needs to be investigated in order to make sure that translators do not bias simulation results.

We have also presented and tested an artificial stock market component based on an orderbook, which implies that quotation is *asynchronous* and *continuous* as on real markets. This is opposed to classical approaches, which aggregates agents decisions synchronously with an equation as a substitute for market interaction mechanism. First results show that it is possible to reproduce most of the *stylized facts* observable on real markets with a pure multi-agents model based on local interactions. This may confirm recent statements implying that most market features are due to the exchange process more than to agents behaviors.

We argue that such continuous and asynchronous models should be used in stock markets simulations. The orderbook model is so close to reality that no validation problems subsist about the mechanism used to make the agents exchange stocks. Moreover, developing agents behaviors is simplified: real traders investment strategies could be implemented "as is", without having to modify their output to match the model requirements.

Concerning technical issues, we can notice that the orderbook does not require specific parameters: this ensures that no hazardous tweaking is necessary to make the market model work in a proper way. Moreover, our model is carefully designed with respect to multi-agents modelization paradigms: by adapting blackboard mechanism and well-known techniques of scheduling to the field of market simulation, we reduce the probability to get unwanted side effects due to technical issues in our simulations.

Now that we both have a realistic market model and a generic market architecture, we are going to be able to compare our model with other ones from the litterature. By doing such intensive experiments, we hope to bring some more elements to the theories which impute most of the stylized facts to the market structure. We will also be able to test new investment strategies coming from classical economic litterature such as the self referential agents proposed in (Orlean 1999).

References

ANDERSEN, J., AND D. SORNETTE (2003): "The $-game," *European Physics Journal*, 31, 141–145.

BAK, P., M. PACZUSKI, AND M. SHUBIK (1996): "Price variations in a stock market with many agents," .

BARON, B. L., W. ARTHUR, AND R. PALMER (1999): "Time Series Properties of an Artificial Stock Market," *Journal of Economic Dynamics and Control*, 23, 1487–1516.

CINCOTTI, S., S. FOCARDI, M. MARCHESI, AND M. RABERTO (2003): "Who wins ? Study of long-run trader survival in an artificial stock market," *Physica A*, 324, 227–233.

CINCOTTI, S., L. PONTA, AND S. PASTORE (2006): "Information-based multi-assets artificial stock market with heterogeneous agents," in *WEHIA06*.

GHOULMIE, F., R. CONT, AND J. NADAL (2005): "Heterogeneity and feedback in an agent-based market model," *Journal of Physics: Condensed Matter*, 17, 1259–1268.

GODE, D. K., AND S. SUNDER (1993): "Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality," *Journal of Political Economy*, 101, 119–137.

ORLEAN, A. (1999): *Le pouvoir de la finance*. Odile Jacobs.

PALMER, R., W. ARTHUR, J. HOLLAND, B. LEBARON, AND P. TAYLER (1994): "Artificial Economic Life: A Simple Model of a Stockmarket," *Physica D*, 75, 264–274.

RABERTO, M., S. CINCOTTI, C. DOSE, S. FOCARDI, AND M. MARCHESI (2005): "Price formation in an artificial market: limit order book versus matching of supply and demand," *Nonlinear Dynamics and Heterogenous Interacting Agents*.

RABERTO, M., S. CINCOTTI, S. FOCARDI, AND M. MARCHESI (2001): "Agent-based simulation of a financial market," *Physica A*, 299, 320–328.

RABERTO, M., S. CINCOTTI, S. FOCARDI, AND M. MARCHESI (2003): "Traders' Long-Run Wealth in an artificial financial market," *Computational Economics*, 22, 255–272.