SUM: a Surprising (Un)realistic Market: Building a Simple Stock Market Structure with Swarm

Pietro Terna

Dipartimento di Scienze economiche e finanziarie G.Prato - Università di Torino
corso Unione Sovietica 218bis
10134, Torino, Italia

pietro.terna@unito.it

ABSTRACT: With SUM, a Surprising (Un)realistic Market, we are dealing with the micro-foundations of a stock market. We avoid any artificially simplified solution about price formation, such as to employ an auctioneer to clear the market; on the contrary, our model produces time series of prices continuously evolving, transaction by transaction. The core of the model is represented by a computational structure that reproduces closely the behavior of the computerized book of a real stock market. The agents send to the book their buy and sell orders, with the related limit prices. The book executes immediately the orders if a counterpart is found in its log; otherwise, it records separately the buy and sell orders, to match them with future orders. The book is cleared at the beginning of each day.

Our (un)realistic market emerges from the behavior of myopic agents that: (i) know only the last executed price, (ii) choose randomly, in a balanced way, the buy or sell side and (iii) fix their limit price by multiplying the previously executed price times a random coefficient. This structure generates increasing and decreasing price sequences with relevant volatility. Also bubbles and crashes appear in this market, generated within the market structure, without the need of exogenous explanations.

Finally, imitation and stop loss behavior are introduced.

KEY WORDS: economic simulation, agent based models, bubbles, Swarm

1. Introduction

According to Gilbert and Terna (2000):

> Ostrom (1988) proposed that there are three different “symbol systems” available to social scientists: the familiar verbal argumentation and mathematics, but also a third way, computer simulation. Computer simulation, or computational modelling, involves representing a model as a computer program. Computer programs can be used to model either quantitative theories or qualitative ones. They are particularly good at modelling processes and although non-linear relationships can generate some methodological problems, there is no difficulty in representing them within a computer program.
>
> The logic of developing models using computer simulation is not very different from the logic used for the more familiar statistical models. In either case, there is some phenomenon that we as researchers want to understand better. This is the “target”. We build a model of the target through a theoretically motivated process of abstraction (this model may be a set of mathematical equations, a statistical equation, such as a regression equation, or a computer program). We then examine the behaviour of the model and compare it with observations of the social world. If the output from the model and the data collected from the social world are sufficiently similar, we use this as evidence in favour of the validity of the model (or use a lack of similarity as evidence for disconfirmation).

The question now is: if our computer simulation model is based upon agents (e.g. built with Swarm[1], as the models presented here), to what extent must our agents be sophisticated? Should we provide them with a “mind”? The answer ranges from the simplicity principle (Axelrod, 1997) to the use of full BDI (Beliefs, Intentions, Desires) cognitive agents.

A possible classification is:

A. “no-minded” agents, that behave in an unstructured environment;

B. learning or “minded” agents, that behave in an unstructured environment;

C. "no minded" agents, operating in a structured environment (our case);

D. learning or “minded” or imitative agents, operating in a structured environment.

In Terna (2000b) we discuss different models with rigid “no-minded” agents that behave in an unstructured market generating cycles and chaos, or with learning “minded” agents, that assure some stability to an emerging unstructured market. Here, in Section 2 and 3, we present "no minded" agents operating in a structured market, with a sophisticated outcome. No generalized results come from this presentation, but many useful suggestions. See Section 4 for a comment about the necessity of integrating the different cases in a unique framework to improve the comparability of the different situations and for further developments of the SUM project.

[1] See <http://www.swarm.org>.

2. The stock market model

With SUM, a Surprising (Un)realistic Market, we are dealing with the micro-foundations of a stock market, employing simple "no minded" agents, but reproducing exactly the rules of a real market. We avoid any artificially simplified solution about price formation, such as to employ an auctioneer to clear the market; on the contrary, our model produces time series of prices continuously evolving, transaction by transaction.

The core of the model is represented by a computational structure that reproduces closely the behavior of the computerized book of a real stock market. The agents send to the book their buy and sell orders, with the related limit prices. The book executes immediately the orders if a counterpart is found in its log; otherwise, it records separately the buy and sell orders, to match them with future orders. The book is cleared at the beginning of each day.

Our (un)realistic market emerges from the behavior of myopic agents that: (i) know only the last executed price, (ii) choose randomly, in a balanced way, the buy or sell side and (iii) fix their limit price by multiplying the previously executed price times a random coefficient. We introduce also the rule of buying with a fixed probability (here $p = 0.5$) if the price falls below a specific floor. This structure generates increasing and decreasing price sequences with relevant volatility. Also bubbles and crashes appear in this market, generated within the market structure, without the need of exogenous explanations.

The emergence of this kind of anomalies in a model of type C (see above) is particularly interesting, because it shows the importance of rules (here the technical structure of the market) in influencing behavior and, mainly, interaction among agents.

So, in the following artificial experiments, simple agents produce complex results. The question is: Are human agents so far from the complexity of the economic system, as ants are from their anthill?

Swarm represents for our task the correct developing framework: it provides a multilayer structure and offers the computational power needed to run the experiments for a sufficient number of cycles. Here the multilayer structure contains: (a) the observer layer, that displays the results, and (b) the model layer, that runs either the time schedule and the environment, with the stock market (realistic) book and the (unrealistic) agents.

3. Our artificial experiments

We introduce here several experiments based on SUM 0.48; you can download it from http://eco83.econ.unito.it/~terna/cef2000pterna/cefpterna.html and run it with Swarm 2.0 or 2.1.

The parameters are the following.

In the Observer (the Swarm side of the program related to the observation of the results) we have:

- *displayFrequency*, the frequency at which the graphic widgets are updated; e.g. if its value is 1000, only one price every 1000 will be reported in the graph;
- *stopAtEpochNumber*, the number of simulated days at which the run will stop; a day is the time required to allow all the agents to make an action; any action is a tick of a clock that makes *agentNumber* (see below) ticks per day.

In the following experiments we have always *displayFrequency* = 1000 and *stopAtEpochNumber* = 2000.

In the Model (the Swarm side of the program related to the execution of the agent based simulation) we have:

- *agentNumber*: the number of agent acting in the model in each day, one per tick (see above);
- *probOfImitatingTheMarket*: the probability that an agent would choose the buy or sell side as an imitation act, i.e. buying if the market mean price is increasing from day -2 to day -1 and the doing contrary if the price is decreasing;
- *probOfLocalImitation*: the probability that an agent would choose the buy or sell side on the basis of the majority of the decisions of the last N agents (here N = 20);
- *asymmetricBuySellProb*: if one of the two strategies described above is adopted, this is the probability *p* of buying or selling as the imitative behavior suggests or of doing the opposite (1-*p*); in absence of an imitative behavior, the probability of choosing the buy or the sell side of the market is 0.5;
- *agentProbToActBeforeOpening*: the probability of placing an order in the opening phase; so a day starts without an empty book, with a realistic effect (anyway, not crucial for the results);
- *minCorrectingCoefficient*: the min value of the random multiplying coefficient *k* used to fix the price of an agent's buy or sell proposal (last price**k*);
- *maxCorrectingCoefficient*: the max value of the previous *k* coefficient;
- *asymmetricRange*: the correction added to the previous min and max limits to adopt an asymmetric behavior, if any (this parameter is not used in this paper);
- *floorP*: the floor price said below;
- *agentProbToActBelowFloorPrice*: the probability that an agent would buy if the price falls below *floorP*;

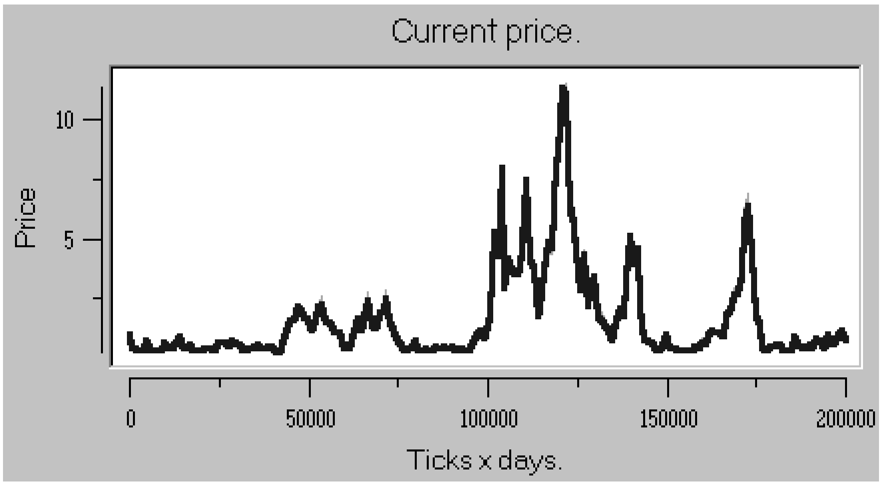

Fig. 1. Only myopic random behavior, first version

- *maxOrderNumber*: the max buying or selling quantity in each order placed by an agent (once a day); the actual quantity *n* is chosen randomly in a range from 1 to *maxOrderNumber*; the program emulate the different quantities in orders repeating *n* times - in the same tick - an order of one unit;
- *meanPriceHistoryLength*: the length of the vector of mean prices;
- *localHistoryLength*: the length of the vector recording agent actions;
- *agentProbToAdoptStopLoss*: the probability that an agent sells or buys to stop loss (we do not account for the real agent situation, i.e. if it is "long" or "short" on the market) if the current price is decreasing or increasing, at a rate greater or equal to the *maxLossRate* parameter, if compared to the mean price of the day *t* - *stopLossInterval*;
- *maxLossRate*: see above;
- *stopLossInterval*: see above.

3.1 *Basic runs*

In the run reported in Fig. 1, 2 and 3, we adopt the hypotheses (i), (ii) and (iii) introduced above (Section 2). The emergence of bubbles and crashes that appear in this framework is a direct the consequence of the structure (the electronic book) of the market.

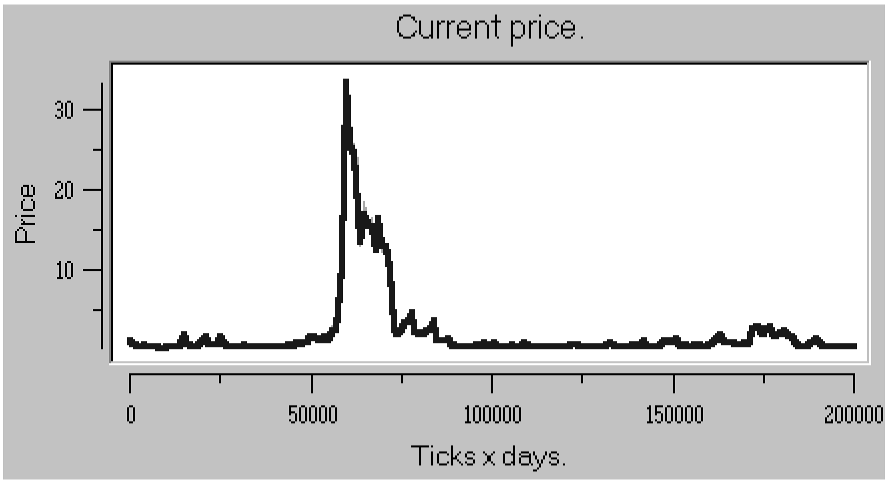

Fig. 2. Only myopic random behavior, second version

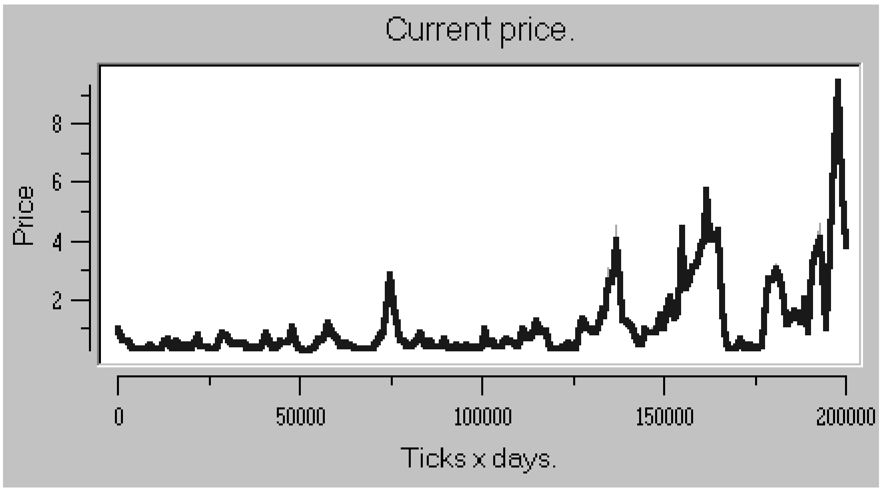

Fig. 3. Only myopic random behavior, third version

Basic parameters are: *agentNumber* = 100; *probOfImitatingTheMarket* = 0; *probOfLocalImitation* = 0; *asymmetricBuySellProb* = 0.9; *agentProbToActBeforeOpening* = 0.05; *minCorrectingCoefficient* = 0.9; *maxCorrectingCoefficient* = 1.1; *asymmetricRange* = 0; *floorP* = 0.3; *agentProbToActBelowFloorPrice* = 0.5; *maxOrderNumber* = 1; *meanPriceHistoryLength* = 100; *localHistoryLength* = 20; *agentProbToAdoptStopLoss* = 0; *maxLossRate* = 0.05 (this is the default value, never used if the previous prob. parameter is 0); *stopLossInterval* = 1 (this is the default value, never used if the previous prob. parameter is 0.

Looking inside the model we can verify that bubbles and crashes emerge mainly from situations in which one side of the market (sell or buy) in empty or near empty.

In Fig. 2 we repeat the artificial experiment with a different random seed (option -s in Swarm), to be sure that the critic result of the bubble appearance is always emerging. Also in Fig. 3 we have the same experiment with a different random seed.

The sequences of prices are reported on a scale measuring the number of ticks time the number of days; so 200000, with 100 agents-ticks, identifies 2000 days of transactions. Prices are generated transaction by transaction, one per tick (if the agent required to act in a tick does not act, the price is kept unchanged). As we have seen introducing the *displayFrequency* parameter, to speed up the execution we display on the graph only one price every one thousand.

3.2 *General imitation (or market imitation)*

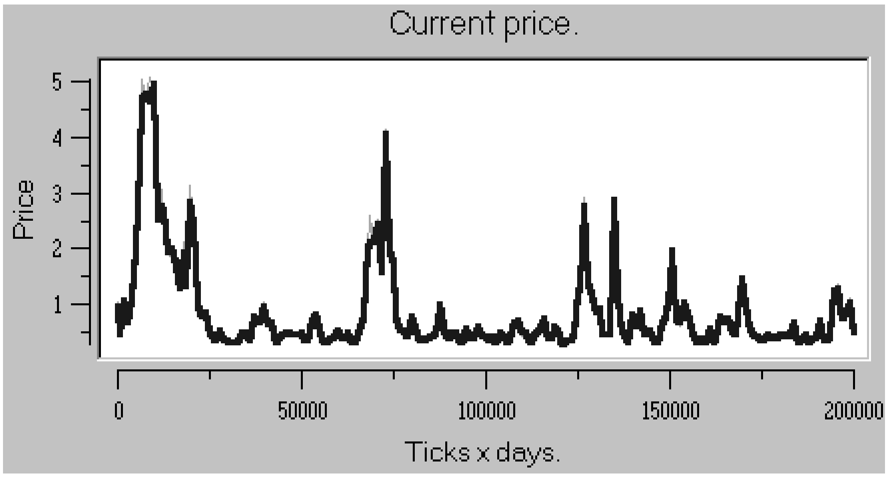

Fig. 4. All parameters of Fig. 1 unchanged, but *probOfImitatingTheMarket* = 0.01

We relax now hypothesis (i) - to know only the last executed price - and (ii) -to choose randomly, in a balanced way, the buy or sell side - for a small quota of the agents, in order to investigate the consequences of the presence either of subjects imitating the market (general imitation) and of subjects locally imitating other agent's behavior (local imitation). Their choice of the operating buy or sell side will be unbalanced, following the *asymmetricBuySellProb* introduced above.

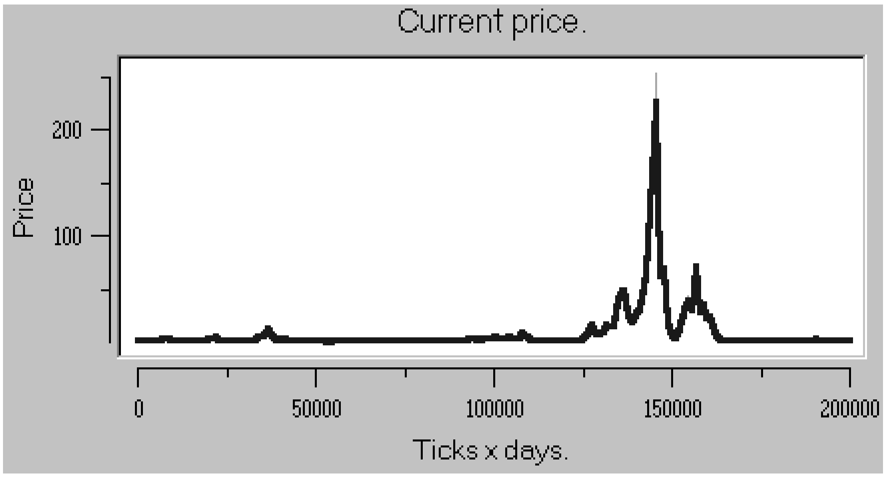

Fig. 5. All parameters of Fig. 1 unchanged, but *probOfImitatingTheMarket* = 0.05

The *probOfImitatingTheMarket* parameter measures the probability that an agent would choose the buy or sell side as an imitative act of the market as a whole, buying with probability *asymmetricBuySellProb* if the mean price is increasing from day *t-2* to day *t-1* and selling with the same probability if the price is decreasing; this is an imitation effect, but also a proxy of the behavior of agents adopting simple technical analysis (chartists). The presence of this kind of agents - also in small quotas - deeply increases the appearance of bubbles and crashes. See Fig. 4 and 5 (where we have an enormous bubble).

3.3 *Local imitation*

In Fig. 6 and 7 each agent - with probability *probOfLocalImitation* - uses *asymmetricBuySellProb* (here 0.9) to buy or sell following the majority of the last N (here 20) other agent's decisions.

Local imitation seems to introduce noise in the results.

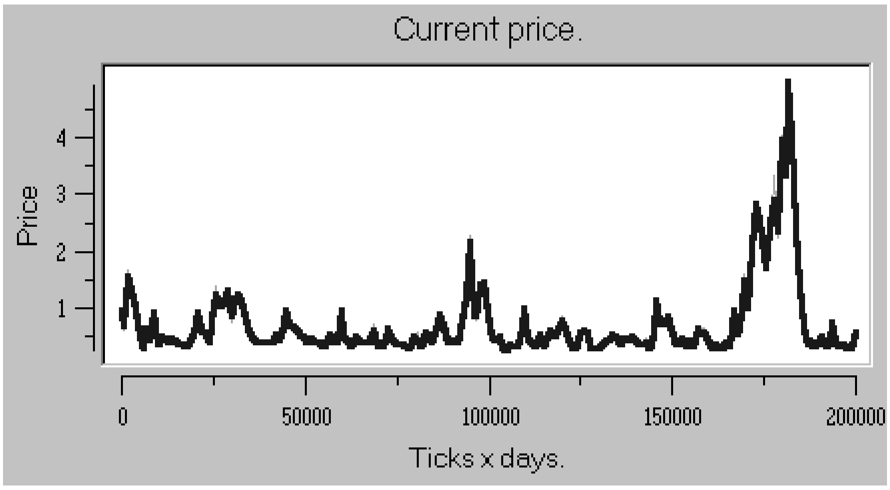

Fig. 6. All parameters of Fig. 1 unchanged, but *probOfLocalImitation* = 0.03

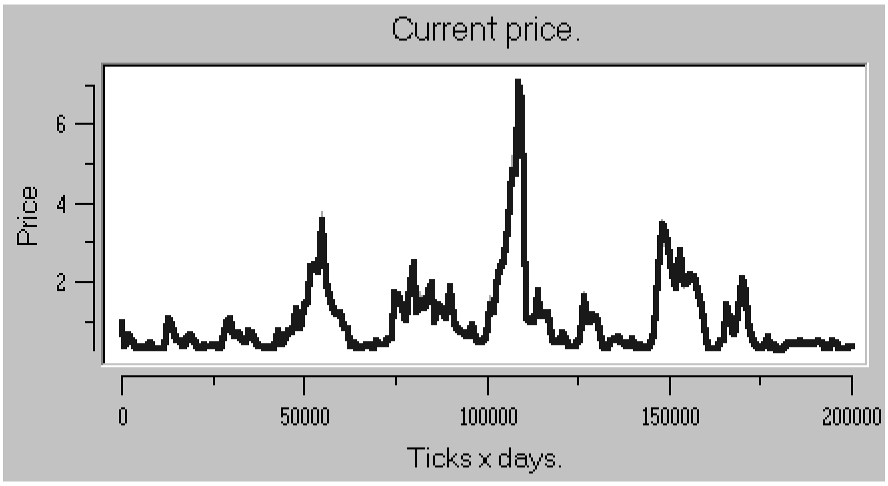

Fig. 7. All parameters of Fig. 1 unchanged, but *probOfLocalImitation* = 0.06

3.4 *Joining global and local imitation*

We join now the two situations above with unexpected highly noisy consequences.

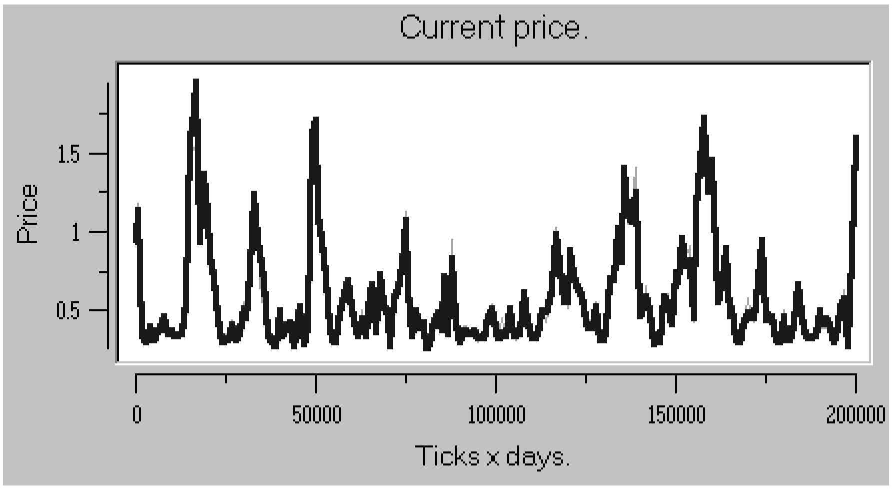

Fig. 8. All parameters of Fig. 1 unchanged, but *probOfImitatingTheMarket* = 0.01 and *probOfLocalImitation* = 0.03

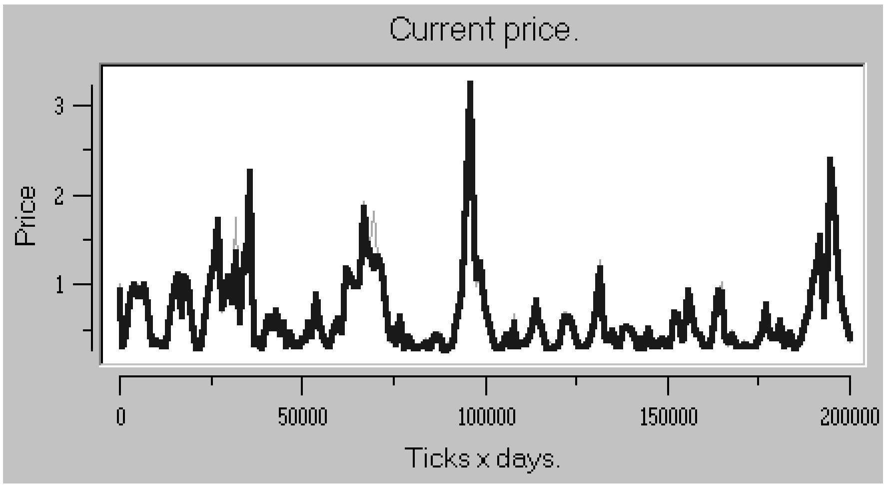

Fig. 9. All parameters of Fig. 1 unchanged, but *probOfImitatingTheMarket* = 0.05 and *probOfLocalImitation* = 0.06

3.5 *Stop loss behavior*

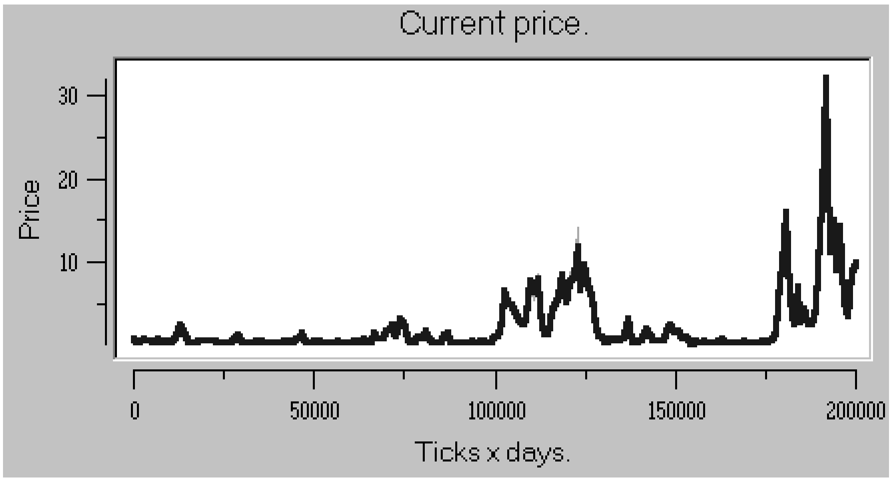

Fig. 10. All parameters of Fig. 1 unchanged, but *agentProbToAdoptStopLoss* = 0.05; *maxLossRate* = 0.10; *stopLossInterval* = 2

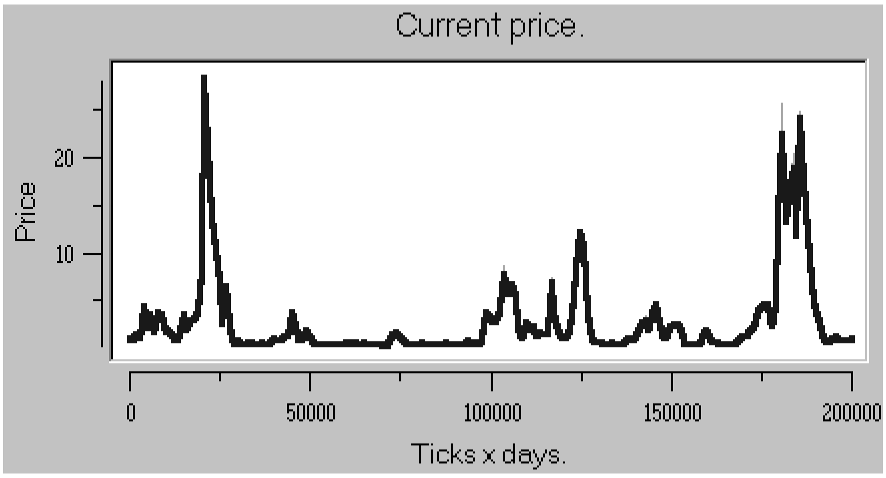

Fig. 11. All parameters of Fig. 1 unchanged, but *agentProbToAdoptStopLoss* = 0.05; *maxLossRate* = 0.10; *stopLossInterval* = 2 and *probOfLocalImitation* = 0.06

Finally, in Fig.10 and 11, we introduce the stop loss behavior (if loss>10% in two days, the 5% of the agents apply stop loss selling or buying if the current price is decreasing or increasing). The effect of stop loss is a heavy one and is amplified by imitation.

3.6 *Many agents*

We also investigate in Fig. 12 and 13 two situations with many agents (here 1000) that would generate a white noise market unless we allow random differences in their buying or selling quantities (in a 1-100 range).

The emergence of bubbles, also in this highly computational time runs, is very important to check the consistency of our results.

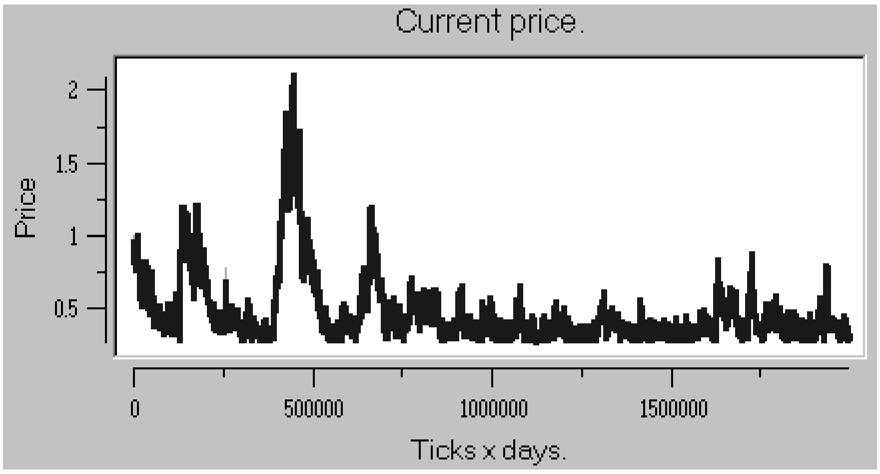

Fig. 12. All parameters of Fig. 1 unchanged, but *agentNumber* = 1000 and *maxOrderNumber* = 100

4. Conclusions and further developments

This structure generates increasing and decreasing price sequences with relevant volatility, bubbles and crashes, as a consequence of the rules of the market.

From the “no-mind” in agents perspective, we show here that it is possible to generate complex patterns without using BDI agents, if the structure of the market is highly sophisticated, and consequently able to generate endogenously sequences of prices linked to the agents’ actions in nonlinear ways.

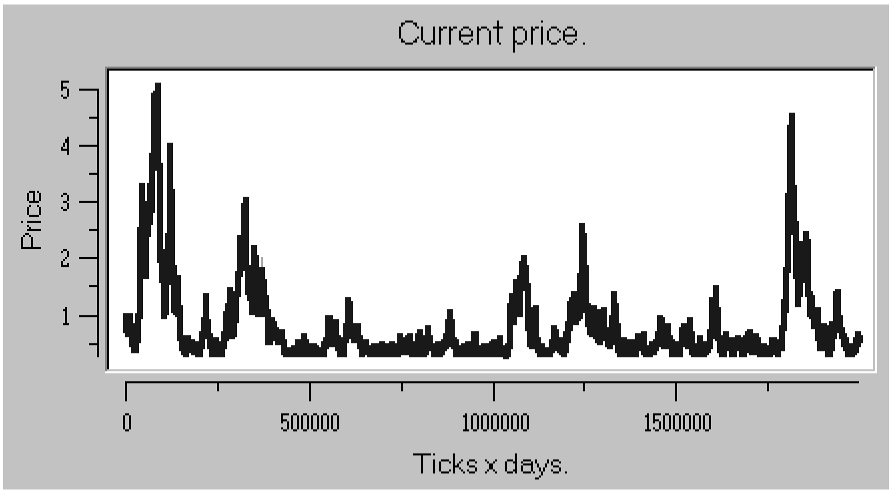

Fig. 13. All parameters of Fig. 1 unchanged, but *agentNumber* = 1000 and *maxOrderNumber* = 100 with *probOfImitatingTheMarket* = 0.05

Further developments may arise, within the framework of Sections 2 and 3, introducing cognitive agents to investigate the consequences of the presence of units able to learn from their experience. In some way, the last ones can correspond to the artificially intelligent agents behaving as econometricians proposed by Sargent (1993), with the interaction of minded agents and structured environments or markets. In doing that we will use the Cross Targets technique introduced in Terna (2000a) to train artificial neural networks to develop minimal behavioral rules.

More generally, we hope that further developing this model we will be able to better investigate empirical puzzles that are hard to understand using the traditional representative agent structure.

Among these puzzles, the time series predictability and the volatility persistence.

Finally, the framework of Sections 2 and 3 is the natural candidate to develop a unified environment, with the goal of comparing directly - in a unique structure - the four extreme situations of (A) no-minded agents behaving in an unstructured environment, (B) minded agents behaving in an unstructured environment, (C) no-minded agents behaving in a structured environment and (D) minded agents behaving in a structured environment.

C is our case here; D will be the first development of this work.

5. References

AXELROD R. (1997), Advancing the Art of Simulation in the Social Sciences, in R.Conte, R.Hegselmann and P.Terna (eds.), *Simulating Social Phenomena*, Lecture Notes in Economics and Mathematical Systems 456, pp.21-40, Springer-Verlag, 1997.

GILBERT N., TERNA P. (2000), How to build and use agent-based models in social science, *Mind & Society*, no. 1, 2000.

OSTROM T. (1988), Computer simulation: the third symbol system. *Journal of Experimental Social Psychology*, vol. 24, 1998, pp.381-392.

SARGENT T.J. (1993), *Bounded Rationality in Macroeconomics*, Oxford, Clarendon Press.

TERNA P. (2000a), Economic Experiments with Swarm: a Neural Network Approach to the Self-Development of Consistency in Agents' Behavior, in F. Luna and B. Stefansson (eds.*), Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*. Dordrecht and London, Kluwer Academic, 2000.

TERNA P. (2000b), *The "mind or no-mind" dilemma in agents behaving in a market,* to be presented to ICCS&SS II, Paris, September 18-20, 2000.