between a ratio of relative increments of CDS spread and equity returns (elasticity) and entity's credit rating.

Finally, we have made an attempt to roughly test performance of the method of determination of margin requirements based on the structural approach. It was applied to CDS written on a Gazprom senior debt denominated in USD. Obtained results have shown that method performed very conservatively, but it may be used as benchmark in terms of potential losses coverage. Meanwhile, the considered method requires a further development and an assessment of its performance in a case of defaulted entities.

Referances

1. Cao, C., Zhong, Z., Yu, F., "The Information Content of Option-Implied Volatility for Credit Default Swap Valuation", FDIC Center for Financial Research, Working Paper. 2009.

2. Finger, C. (et.). CreditGradesTM technical document, RiskMetircs Group. 2002.

3. Hagan, Patrick S., Kumar, D., Lesniewski, A. S., and Woodward, D. E.: "Managing smile risk", Wilmott Magazine, 84-108. 2002.

4. Raunig, B. , Scheicher, M. "A Value at Risk Analysis of Credit Default Swaps", ECB Working Paper No. 968 . 2008.

5. Tang, D.Y., Yan, H., "Market Conditions, Default Risk and Credit Spreads," Journal of Banking & Finance, 34, p.743-753. 2010.

FINANCIAL MARKET SIMULATIONS

P. Kumar, S. Ivliev

Prognoz Risk Lab, Perm

Agent-based simulation has emerged as a promising research alternative to the traditional analytical approach in studying financial markets. The idea behind agent-based simulation is to constructively model a market in a bottom-up fashion using artificial agents. Under the umbrella of Market Microstructure Project (MMP), a Prognoz Risk Lab initiative, financial market simulations is a major attempt to construct an agent based *model of stock market with a very large population of autonomous, purposive agents interacting in a complicated economic environment. To build it, major advances are needed in terms of market microstructure modeling and software engineering. In this paper, we describe general structure of stock market, in particular continuous double auctions developed for MMP and present financial market simulations that will be used to describe the artificial autonomous agent having intelligence. It has ability to run the models on the massively parallel clusters. The idea is simple enough to build up models which replicate the highly stylized facts of stock market behavior. The overall exercise will help us to understand the stock market and trader's behavior to arrive at some regulatory issues, if there is any.*

1. Introduction

With the advance of computing power and modeling tools, computer simulations have become increasingly important in many financial applications, such as derivative pricing and risk management. Now the already-booming family of financial simulations is witnessing the rise of a new member: agent-based market simulation. Since the beginning of the 1990's, there has been a surge of interest within the finance community (both in industry and academia) in employing agent-based methods to obtain insights about market microstructure and to experiment with trading strategies. Research in agent-based financial markets naturally provides an excellent opportunity for interdisciplinary collaborations, because it often involves joint projects involving financial engineers, economists, computer scientists, mathematicians, statisticians, physicists, etc. In this paper, we will present our agent based architecture that is specifically designed for agent-based modelling and simulation of continuous double auction markets (CDA).

2. Literature Survey

2.1 Agent based financial markets

The interest devoted to agent-based design of financial markets in the last twenty years is testified by a large literature on the subject. Many of these studies share a number of common features. They generally concentrate on agents' behavior, comparing different trading strategies (Chiarella and Iori 2001; Hommes 2002; Cincotti et al. 2003; Raberto et al. 2003; LiCalzi and Pellizzari 2003), and implement some learning mechanisms covering different hypotheses such as zero-intelligence random traders (Raberto et al. 2001), genetic algorithms (Arifovic 1996), or neural networks (Arthur et al. 1997). The main goal of these studies is to reproduce the evolution of market prices and to capture some empirical stylized facts such as volatility clustering of price returns or fat tails in the distribution of returns (see LeBaron 2006; Samanidou et al. in press for recent surveys).

2.2 Computational technique

The urge to understand complex dynamism of financial market through agent based modelling and simulation demands highly sophisticated computationally technique having its origins in discrete event simulation, genetic algorithms and cellular automata. ABMs have been adapted to run on novel architectures such as GPGPU, e.g. nVidia hardware using the CUDA programming language. Argonne National Laboratory has a Web site on Exascale ABMs and has run models on the IBM BlueGene with funding from the SciDAC Programme.

For a list of ABMs software, some highly specialized; (see; Leigh Tesfatsion's web site http://www.econ.iastate.edu/tesfatsi/acecode.htm.) Basic documentation for most of the packages reviewed is largely incomplete. As an example, the open source, Swarm communities have a web site and wiki which contains a lot of information but is hard to navigate. The extended list of ABMs, software engineering part is well enlisted in survey (Allan, 2010).

Flexible Agent Modeling Environment (FLAME)[19] has been developed to allow a wide range of agent and non-agent models to be brought together within one simulation environment. It is aimed principally at the medical and biological domains, for example studies of tissue cultures and signaling pathways. It was used in the EU funded EURACE project for financial modeling.

3. Objective and Contribution

The research described in this paper can be viewed as an attempt to bridge the gap between the relevant disciplines (such as finance, computer science, mathematics, statistics, physics, etc.) involved in the field of agent-based computational finance.

What is important in the MMP project, and has been scarcely investigated in the field of agent-based modeling, is the interaction between

---

[19] FLAME is being developed primarily at the University of Shefield, see http://www.flame.ac.uk and http://www.flamegpu.com with collaborators at STFC.

the agents and the embedding real economic system using data from The Moscow Interbank Currency Exchange (MICEX).

This paper contributes a high-quality software platform that is specifically designed to carry out agent-based simulations of financial markets. The platform is flexible enough to accommodate many different kinds of agent-based artificial market models

The paper also present stylized facts captured by various academician like Smith, E., J. D. Farmer, L. Gillemot, and S. Krishnamurthy (Smith, 2003).

4. Financial Model

4.1 Continuous double auction market

Market trading is governed by a market mechanism. Such mechanisms are designed to define the exchange process between buyers and sellers, by specifying the set of messages that can be exchanged (e.g. the traders' actions such as submitting a bid or an ask, or agreeing to a transaction) and by specifying the resource allocation process given the received messages (e.g. when transactions occur given the exchanged messages and at what price these transactions occur).

In the continuous double auctions (CDA), there is usually a fixed-duration trading period (typically referred to as a trading day), and buyers and sellers can submit bids and asks, respectively, at any time during the trading day and the market clears continuously. Specifically, the market clears whenever there is a match between open bids and asks (i.e. a transaction is possible). In a single-unit, single-attribute CDA, the market clears (with a single trade) whenever the outstanding bid is at least as high as the outstanding ask. All messages submitted by traders are usually public and announced to all the participants in the market.

There are two aspects that characterise the CDA; namely the structural and the behavioural. The structural perspective is determined by the market protocol which is a set of interaction rules and a set of clearing and pricing rules. The interaction rules define how participants interact through a set of actions. The behaviour is what emerges from the interactions of the buyers and the sellers in the market and, it depends on the strategies of all the agents in the market. In this context, the agents strategies within the given market mechanism to determine what actions they should take, at what time.

However, the complex nature for CDA discards the concept of game-theoretical analysis (Gode, 1992), which assume the concept of dominant strategy producing high profit in the auctions. Thus, over the past decade, there has been considerable research endeavour in developing trading strategies that define how agents should behave based on a variety of heuristic approaches. We have used the similar approach where agents are heterogeneous in terms of their trading strategies. We use different technical indicators or combinations of the same, evolving over period of time using evolutionary algorithms like genetic programming. In this paper, we will be using our in-house CDA engine, having all the features enlisted above, with its own price formation mechanism depicting the real market. It doesn't completely assume the equilibrium price formed after intersection of demand and supply curve.

5. Simulation Platform

Our simulation platform is a distributed (client/server architecture) and multi-threaded computing environment that is capable of simulating a variety of financial markets for different research purposes. Central to the simulation platform is an artificial market (implemented as the server) which takes requests from market participants (implemented as clients) and processes them according to rules that are consistent with the type of the artificial market. The artificial market currently implemented on the simulation platform is a continuous double-auction market.

The whole objective to throughout the development process of simulations platform is to have scalability; the ease with which a system or component can be modified to fit the problem area. The next is achieving high performance of the server. In other words, we want the server (market) to be able to process incoming requests and generate responses at a very fast pace.

5.1 Trading platform

We tend to use Flexible Large-scale Agent Modelling Environment (FLAME) to host our CDA engine modelled in DELPHI, which sits on C/C++ compiler. By using FLAME modellers can easily create, exchange, include and couple models that have been written in a high-level modelling language. Other key aims were the development of parallelization techniques, the distribution of agents over many processors, and the inclusion of testing methods to verify developed models.

FLAME is based on so-called finite-state machines that are, on automata described by a finite number of states, transitions between those states, and actions, that are heavily used in computational sciences (Coakly, 2007). The approach taken in FLAME is to regard each individual agent as a X-Machine and to specify a communication structure such that the different agents can exchange messages with each other. In other words, individual X-Machines are given the ability to communicate by exchanging messages. Moreover, they are generalized by providing them with an internal memory, leading to a so-called stream X-Machine design. The FLAME framework provides the modellers flexibility to write the models in any language sitting on C/C++ compiler and adhered to the norms of FLAME simulation framework coded in XML for dynamism. For in details description of the FLAME framework, see (Deissenberg, 2007).

5.2 Agent design

Trading agents are heterogeneous in nature in existing agent based models. Every agent has their own different trading strategy, and they have learning ability as discussed below.

5.3 Learning

When we consider learning on the asset market we want to let the agents search for the best trading rules, which implies a search through a possibly very large space. The rule space that agents must search through to find better performing rules consists of the space of parameter settings for the different rule classes. The search space may be vast since it depends not only on the number of parameters in each class of rules but also on the discretization of the parameter space.

For this a GA is by far the best option, since it includes experimentation and exploration, and recombines rules that have performed well in the past to form new rules that may perform even better in the future. However, due to the co-evolution of the rule population, rules that are well-adapted under certain circumstances may not be faring too well under others, so therefore we keep track of the rule performances in a classifier system.

Rule performance can be measured using different performance measures, possibly affecting the results of the learning dynamics. We have selected as a performance measure the capital gains that the rule has produced, averaged over the holding period of the portfolio that was suggested by the rule. This seems a natural measure for a rule's performance since it matches the objective of the financial traders, which is to obtain as large a capital gain as possible on their investments.

We divide learning into three components, a Learning Classifier System (LCS) to register the performance of a population of rules, a Genetic Algorithm (GA) to modify the population of rules, and Experience Weighted Attraction (EWA) learning to update the rule attractions and select among them using a probabilistic discrete choice framework.

5.3.1 EWA learning mechanism

We include a learning algorithm on the asset market by using the EWA learning mechanism (Pouget 2007). EWA learning incorporates two learning effects, namely the law of actual effect and the law of simulated effect. The law of actual effect refers to the fact that agents learn from information about the own choices in the past. Selected strategies that were successful in the past will have a higher probability to be selected in the future.

The law of simulated effect refers to the notion that agents learn from information about others' choices in the past. The agent observes (i.e. simulates) the payoffs from non-selected strategies and reinforces the successful ones. For, details review about the mathematical understating please refer (Pouget, 2007).

5.3.2 Classifier system

A standard Classifier System (CS) is a set of classifier rules. A classifier rule consists of a Condition, an Action, and other parameters, such as the Performance (or Strength or Accuracy) of the rules. The 'bid' of a rule is then usually defined as the rule-strength plus a small noise term. In our approach, we will use a discrete choice- or multinomial logit model to select the rules (Deissenberg, 2007).

5.3.3 LCS

In comparison to regular Classifier Systems, a Learning Classifier System (LCS) has the additional property that new rules can be created and old rules can go extinct (but the population size remains fixed). This is done through the evolutionary operators of crossover and mutation. There is set of internal and external LCS, to register performance of the rule at the central and local level.

5.3.4 GA mechanism

An efficient sampling method of the space of parameter settings can be obtained using a Genetic Algorithm (GA). However, we have to define how the crossover and mutation operators modify the parameter settings in the rule population.

The pseudo code and learning mechanism is explained in detailed with the help of following algorithms (Fig. 1 and Fig. 2). In learning mechanism (Fig. 3), The updating of the rule set in the Learning Classifier System occurs at the system level, through the application of a Genetic Algorithm that modifies the rules' parameter settings. The EWA learning is occurring at the agent level and concerns the selection of rules on the basis of the performances registered in the external, system level Learning Classifier System. The internal Learning Classifier System of the agent transforms system-level performances into agent-level attractions, which implicitly includes a finite memory of past performances.

```
Algorithm        LEARNING ALGORITHM

function INITIALIZATION
    for Learning Classifier System do                                  ▷ LCS
        create a random population of N_pop bitstrings
        set all performances to zero
        set all user counters to zero
    end for
    for EWA learning do
        set experience to N(0) = 1
        set all attractions to A(0) = 0
    end for
end function

function MAIN
    use current rule in the market                      ▷ Market environment
    for every T_LCS periods do
        report performance to LCS
        run LCS routine
        run EWA learning
    end for
end function

function LCS                                        ▷ Learning Classifier System
    update all rule performances
    every T_GA periods run GA routine
end function

function EWA LEARNING                                         ▷ EWA learning
    read all updated rule performances from LCS
    compute attractions for all current rules in LCS
    for Multi-logit rule do                                ▷ Multi-logit rule
        read attractions
        compute choice probabilities
        select rule at random according to choice prob.
    end for
end function
```

Figure 1: Pseudocode for learning algorithm

```
Algorithm    LEARNING ALGORITHM

function GA                                                          ▷ Genetic Algorithm
    read current generation of bitstrings from LCS
    read current fitness of bitstrings from LCS
    Selection/Reproduction
    Cross-over
    Mutation
    Election
end function
function SELECTION/REPRODUCTION                                      ▷ Reproduction
    create fitness based probabilities
    draw $2N_{rep}$ random copies from the LCS using fitness based probabilities
    ($2N_{rep}$ can be a fixed percentage of the pop. size $N_{pop}$)
    create $N_{rep}$ parent pairs by random matching
    (drawing is with replacement using uniform probabilities)
end function
function CROSS-OVER                                                  ▷ Cross-over
    draw random cross-over point between $[1, L-1]$
    for Parent pair $1 : N_{rep}$ do
        if $rand > p_{cross}$ then
            perform single-point cross-over between parent pair
            add 2 offspings to potential new generation
        else
            2 offspring bitstrings are identical copies of parents
        end if
    end for
end function
function MUTATION                                                    ▷ Mutation
    for Potential new bitstrings $1 : 2N_{rep}$ do
        if $rand > p_{mut}$ then
            draw random mutation
            apply mutation to the bitstring
        end if
    end for
end function
function ELECTION                                                    ▷ Election
    for Potential new bitstrings $1 : 2N_{rep}$ do
        test for higher fitness between 2 offspring and 2 parents
        add 2 out of 4 best bitstrings to new generation
    end for
end function
```

Figure 2: Pseudocode for learning algorithm (contd.)

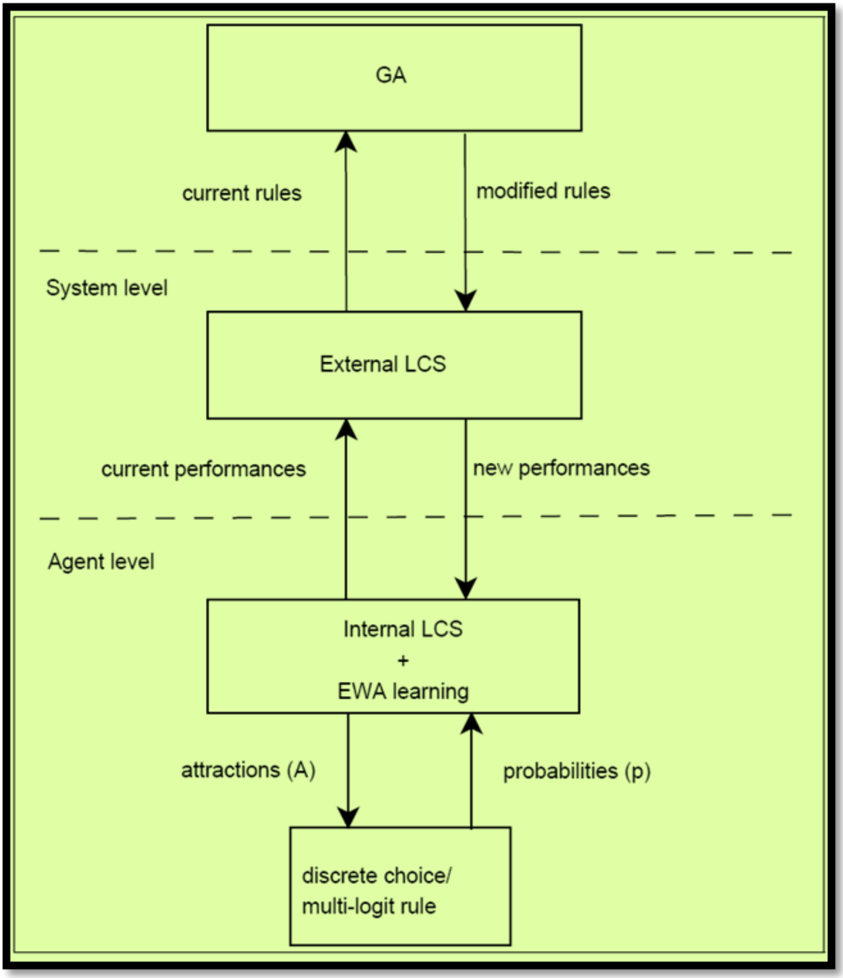

Figure 3: Diagram of the learning mechanism

5.4 Server architecture

The typical server side structure with the working mechanism is as soon in the Figure 4. After the server is started, the main process will listen and monitor the main socket port for incoming agent connections. The read thread is mainly responsible for fetching the agent's requests from the peer socket port, pre-processing the request and pushing pre-processed request to the central server for the queries. The write thread is responsible for writing responses (from the server) to the agent.

The artificial agent structure design is due to FLAME architecture, which gives flexibility to have dynamic memory at the agent level and donot requires any temporary storage in the server. Even the communication between the agent is done through stream X-machines., detailed is discussed above.

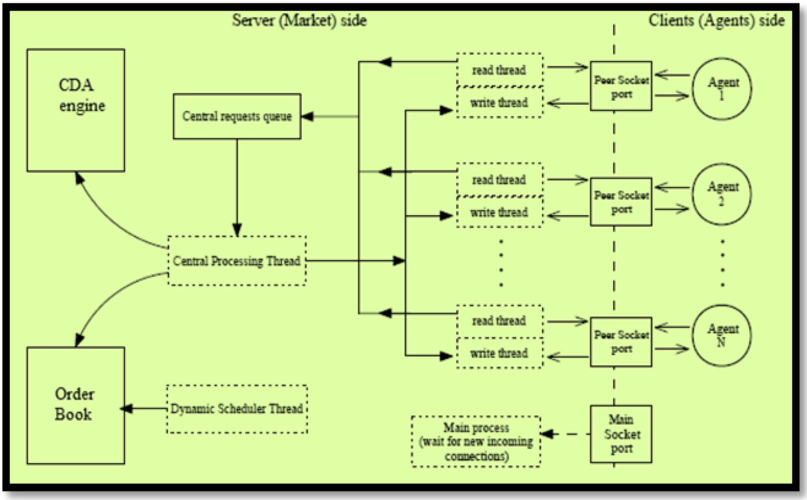

Figure 4: Server structure diagram. Shapes with solid borders on the server side represent essential data structures residing in computer memory and shapes with dashed borders represent major processes/threads running on the server. A link between a thread and a data structure (represented by a thick arrow) suggests that the thread can access and modify the content of the data structure.

6. Experiments

We consider a discrete-time simulator for CDA model, and at each time step, an agent is randomly triggered to submit a bid or an in the market. In line with previous work, we impose a deadline on the duration of the trading day with the auctions closing on 2880 seconds (8*60*60), considering trading occurs for 8 hours. At the beginning of a trading day, buyers and sellers are endowed with a set of limit prices that correspond to goods to buy and sell respectively.

Also, for the controlled experiments, we specify different market setups, say M1, M2, M3,… and some market shocks based on the price to compare the behaviour of the agents.

The market is populated by sets of different buyers and seller, trying to trade in the different market regime using heterogonous trading strategy evolving over period of time by learning from the market and their trading. Each agent is endowed with the some cash for trading in the market. There will be penalty for losses they will make, and some transactions costs.

7. Conclusions

We are still on very development stage, which is giving lot of learning on the economic models and software engineering side. The most important part of agent based models is agent behavior, which we hope can be well understood with our present existing learning architecture. Next, important task is communication between agent and the environment. This is one computationally challenging part, where two times has to be modeled; internal time of agent and main time at the center. This will ease the communication of environment in case of any crash, stop loss ect to the agent. This is only possible if we have synchronous parallel architecture, which is well supported by FLAME.

Reference

1. *Arifovic, J.* The behavior of the exchange rate in the genetic algorithm and experimental economies. J Polit Econ 104 (3), 510–541.1996.
2. *Arthur, W. B., Holland, J. H., LeBaron, B. D., PalmerR., Tayler, P.,* 1997. Asset pricing under endogenous expectations in an artificial stock market. In: Arthur, W., Durlauf, S., Lane, D. (Eds.), The Economy as an Evolving Complex System II. SFI Studies in the Sciences of Complexity. Addison Wesley Longman.
3. *Chiarella, C., Iori, G.,* 2001. A simulation analysis of the microstructure of double auction markets. Quantitative Finance 2 (5), 346–353.
4. *Cincotti, S., Focardi, S. M., Marchesi, M., Raberto, M.,* 2003. Who wins? study of long-run trader survival in an artificial stock market. Physica A 324 (1-2), 227–233.
5. *Deissenberg, C., van der Hoog, S.,* 2007. Modelling requirements for EURACE. EURACE Report D2.1, GREQAM, Université de la Méditerranée

6. *Hommes, C. H.,* 2002. Modeling the stylized facts in finance through simple nonlinear adaptive systems. P Natl Acad Sci USA 99 (Suppl. 3), 7221–7228.
7. *LeBaron, B. D.,* 2006. Agent-based computational finance. Vol. 2 of Handbook of Compu- tational Economics. North Holland.
8. *LiCalzi, M., Pellizzari, P.,* 2003. Fundamentalists clashing over the book: a study of order- driven stock markets. Quantitative Finance 3 (6), 470–480.
9. *Raberto, M., Cincotti, S., Focardi, S. M., Marchesi,* M., 2001. Agent-based simulation of a financial market. Physica A 219, 319–327.
10. *Samanidou, E., Zschischang, E., Stauffer, D., Lux, T.,* in press. Microscopic models of financial markets. In: Schweitzer, F. (Ed.), Microscopic Models of Economic Dynamics.
11. *Smith, E., J. D. Farmer, L. Gillemot, and S. Krishnamurthy*. 2003. Statistical theory of the continuous double auction. Quant. Finance 3, 481-514.
12. *S. Coakley*, 2007. Formal Software Architecture for Agent-based Modelling in Biology, Ph.D. Thesis, University of Sheffield.
13. *Pouget, S.,* 2007. Adaptive traders and the design of financial markets. Journal of Finance (forthcoming).