A Stock Trading System using Genetic Approach and Object-Oriented Database Technology

Jerzy Korczak, Pascal Kustner

Université Louis Pasteur, ERTI Illkirch, France
<jjk, kustner>@dpt-info.u-strasbg.fr

Abstract: *In this paper, a multi-access database system for stock trading using AI-based methods is presented. Financial time series as well as stock trading experts are stored in a relational database. These experts are used to generate trading decisions. A stock trading expert is considered as a composition of technical indicators and financial decision rules that can be used to predict financial time series. Before storing experts in the database, their trading rules are discovered by a genetic algorithm. The goal of the evolution is to find out the most relevant and efficient set of decision rules. Users can download information from the database and carry out trading simulations and take expert trading advice. The proposed methodology is validated on real-life stock data. A few snapshots of the user interface illustrate the approach and results obtained. Based on the results of experimentation, it seems that the genetic algorithms are well adapted to solve these types of problems and allow interesting solutions to be discovered.*

Key words: intelligent trading systems, financial databases, genetic algorithms.

1 Introduction

Investors and financial experts have always focused a great deal of attention on the problem of financial price forecasting. Over the years, financial experts have developed hundreds of technical and fundamental indicators to predict financial time series. Today, financial markets are considered as complex, time-variant and nonlinear dynamic systems.

The problem of stock trading is particularly complex. The basic question is how to select from the large number of features and indicators available those that are most significant and relevant. Although many propositions of stock trading models exist [1,4,7,10,15], until now no efficient solution has been found. Many experts take the view that the model should be looked for in historical data, and the search process should have a continuous and incremental character depending on the efficiency of the currently used model. In practice, this means that at a given moment of the financial time series, it should be possible to design a model that provides a better forecast than any other model in terms of the evaluation function.

The goal of this project is to develop a stock trading support system using database technology and evolutionary computing. In order to introduce the idea of an evolutionary approach, let us assume that each combination of decision rules and indicators represent an artificial trading expert. The process of expert generation is based on the assumption that all the information necessary to predict the trading positions is contained in the financial time series. Experts are not given *a priori* but evolve dynamically through the genetic algorithm. In genetic terminology, the artificial trading experts form the population of chromosomes, where each one generates a trading decision based on his rules applied to historical financial data. The decisions have to be made on a day-to-day basis according to price and volume data. At a given time t, each expert has to take one of three decisions: "Buy", "Sell" or "Hold". In order to measure the expert performance, the prediction horizon of one day is chosen. This means that an expert has to take long, short or neutral positions in the market, each of which incurs a fixed transaction cost of the price at time $(t+1)$.

In our approach, trading experts are dynamically modified by the genetic operations and evolve over time. The rules of evolution require that the profit making experts have to be promoted much more strongly than those taking the wrong decisions and losing. The evolution process has been detailed in our previous works [11,12]. In this paper, we concentrate on the architectural and technological aspects of a multi-access database system for stock trading using genetic methods.

The system makes use of the latest developments in e-commerce technology, such as multi-tier architecture, multi-access database and object-oriented packages written in Java. These techniques allow the system to be used on many platforms and ensure that it is possible for every computer connected on the Web to reach the database. The database technology guarantees not only data coherence but also assures easy access to the data and short response times. In comparison with the other trading systems, iBE provides a trading expertise well-adopted to changing market situations and offers efficient technology allowing remote trading.

In Section 2, the architecture of the decision support system, called iBE, is briefly described as well as its main functions. In Section 3, the expert encoding technique and the decision rules are defined. Here it is shown how trading experts are generated using the financial time series. In Section 4, database management problems are discussed and the programming environment used to develop the system is described. The proposed methodology is illustrated using the time series extracted from the Paris Stock Exchange database. In Section 5 a few snapshots of the trading system show the user interface and results obtained. Conclusions are drawn in the last section.

2 System architecture and functions

The objective of this project was to develop a decision support system for stock trading that integrates the evolutionary approach to data mining and recent information technology, in particular, internet, e-commerce, portables such as WAP, GPRS or UMTS, PDA, handheld computers and portables.

2.1 System architecture

The architecture of the new system is composed of three functional modules (Figure 1). The goal of the first module is to discover by genetic evolution an artificial expert that is capable of advising on optimal trading decisions based on financial time series. In terms of the iBE system, the optimal decision maximizes the function-criterion that evaluates the performances of the expert.

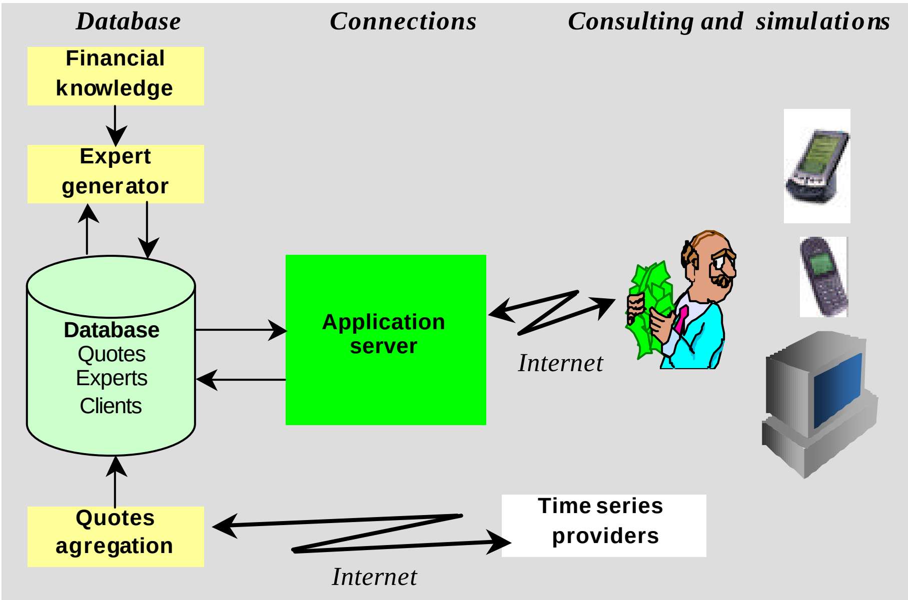

Figure 1. General schema of the system architecture

The financial knowledge of the system corresponds to technical indicators and rules of financial analysis. In our application each rule is represented as a conditional form. Each generated expert is a combination of rules, encoded as a sequence of genes, i.e., a chromosome. Usually, the goal of genetic evolution is to discover the most profitable expert. In our system, various performance measures are available: Return on Investment (ROI), the quality of prediction, the level of risk and the maximum of successive losses

In practice, the goal of the expert generator is to find an expert that outperforms the acceptable level of the function evaluation. To achieve this task, the genetic algorithm must efficiently traverse a very large space of solutions while simultaneously promoting the better experts and eliminating the worst ones. This process will be detailed in the next section.

The second module of the iBE system is the application server, implemented in Oracle8i technology coupled with object-oriented programming. The database contains the generated experts as well as all data required to generate trading advice, in particular, for each stock, the opening, highest, lowest and closing prices, the volume of transactions, and the market index.

2.2 System functions

The iBE system is designed for traders equipped with a PC (standard or portable), and to the trader-nomades that use a portable phone WAP/GPRS/UMTS or the PDA. The information service is diversified according to the visualisation and computing capacity of the tool used. A customer wishing to consult the financial time series in order to take a decision regarding the market can connect to the database, and after his authentication can obtain an advice, an explanation and complementary financial information. It is evident that the third module "*Consulting and simulation*" is essential for users. This module is composed of four principal functions:

- The first function provides the downloading of time series, experts and rule parameters from the database.
- The second function visualises time series, trading rules and decisions taken by each of the rules.
- The third consists of a generation of trading decisions (Buy, Sell or Hold) for the following day. The iBE knows the rules to use and their parameters. Therefore, before seeking advice, the user can inform himself about applied rules and expert performances in a chosen period. The user can equally simulate his own strategy, parallel to that of the expert, and compare his own results to those of the expert.
- The fourth function facilitates the search for financial information available on the WEB taking into account the user's key-words and preferences.

2.3 Important technical choices

In order to implement this system, the latest information technologies have been used to maintain the data coherence, the expertise diffusion and how application will evolve. The system interface is designed for stock-traders who are not necessarily specialists in time-series modelling and computer science. The necessary trading information is available and displayed by a few clicks of the mouse. The user may visualize stock quotes, expert performances and evaluate his and expert trading strategies. The user confidence in the expert decisions is reinforced by the display of the various expert performances measures. To extend this information, the on-line Web-searching services are provided.

The choice of the programming language and the development environment were conditioned by the same requirements as were made on the database management system. The Java language, proposed by SUN, is the language assuring the system portability on different platforms and operating systems, straightforward integration with the database, and easy system extensions in the future.

3 Decision rules and trading expert generation

As mentioned in the introduction, stock market forecasting is a particularly complex task. In order to create a model of expert, we are faced not only with the problem of discovering which composition of decision rules is most efficient, but also of how to estimate the significance of changes over time, and how to take them into account when refining the expert knowledge. The assumption is that the model is hidden in historical data and in data relations that reflect the behaviour of actors within the financial market. Their behaviour is the result of their perception and evaluation of economic and political situations related to a given financial product. It is of course impossible to identify all data, rules and indicators that influence each particular actor.

The goal of the evolutionary approach is to discover an artificial expert which is able to make optimal decisions based on the available financial data. One can assume that the optimal decision is the one which maximizes the relative difference between profit and loss. In this case, the genetic algorithm supervises trading and keeps track of profits for each individual expert as follows:

$$Gain(expert) = Cumulated\_Sell - Cumulated\_Buy - Transaction\_cost$$

To calculate the gain, the initial investment is given by the user with a fixed commission charged when a trade is opened and again when it is closed. Based on the selected financial rules, day after day each expert makes a choice of one of three possible positions: “Buy”, “Sell” or “Hold”. The system computes the gain using the next day opening price for the buy and sell positions. All daily decisions contribute to the global gain of the expert. In addition to the trading gain, the system also calculates the following: the return on investment, which is the amount of money it made on the learning and testing sets, the percentage of correct decisions, the level of risk, and the optimal decision. In the current version of the system, a user selecting from the list of available criteria may choose the model performance measure.

The design of decision rules represents crucial tasks in the evolution process. In our experiments, the financial time series were extracted from the Paris Stock Exchange database.

The decision rules correspond to the financial rules used in methods based on the technical analysis [2,3,6,13]. The background knowledge of the system is represented as a set of production rules stored in the following format:

IF conditions are satisfied THEN decision.

The left part of the rule is written as formal expressions, describing mathematical formulas or financial indicators, but the right part represents the decision that might be taken.

Production rules may be very complex and contain many nested sub-rules and functions. Below, a few examples of production rules are given and commented.

- R1: IF Rate of Change is greater than 1 THEN Buy,
OTHERWISE IF Rate of Change is smaller than 1 THEN Sell

The Rate Of Change (ROC) is a simple relation between the current price and the price in the past; in the system the price is taken 10, 15, 30 and 50 days before.

- R2: IF EMV indicator is positive THEN Buy
ELSE IF EMV indicator is negative THEN Sell

The *ARM's Ease of Mouvement Value* is a momentum indicator used to quantify volume and price changes in one indicator in order to determine the ease, or lack thereof, with which the market price is able to move up or down. The indicator is computed as follows:

$$EMV= [(H+L)/2 - (H_p+L_p)/2]/[V/(H-L)]$$

where H is the highest price in a given period, L is the lowest price, H_p and L_p are the highest and the lowest prices respectively in the previous period, and V is the volume of transactions.

Most of the decision rules in our system have come from the *Encyclopedia of Technical Market Indicators* by W.Colby and T. Meyers [6]; such as rules based on CCI, OBV, etc. Of course, if one wants to create more sophisticated rules, the knowledge base may easily be modified and extended. Rules might even be irrational, for instance a rule which prohibits making any transactions on Friday 13th, or rules based on the signs of the Zodiac.

Each artificial trading expert is encoded as a binary string, a chromosome, made up of a finite sequence of genes. Genes represent the rules to be executed while examining financial data. In our representation, it was assumed that each gene represents one decision rule. For instance, the model using the rules R2 and R4 is encoded as a string: 010100.

As described in the previous section, decisions are generated according to satisfied rule conditions. In our system the following decisions are possible:

- 0 – indicates the "Sell" position, where the system predicts that the stock price will go down,
- 0.5 – indicates the neutral position, "Hold": the stock price neither indicates a "Sell" or a "Buy" decision,
- 1 – indicates the "Buy" position, the system predicts that the stock price will go up.

The decisions are evaluated by the function-criterion that measures the expert performance in a given period of time. Returning to the evolution process, it is desirable that experts with better performance be promoted so that they will be inserted in the next population, but those who are losing be reduced or eliminated from the process. Expert evolution is detailed in [11].

In our system, the model performance measure can be chosen by a user selecting from the list of available criteria: the return on investment, the forecasting accuracy; the level of risk, the maximum consecutive loss. After evaluation, the system sorts the models according to the values of the fitness function. The task of the genetic algorithm is to discover a model that exceeds the acceptable level of the function [8]. To solve this problem, the genetic algorithm must efficiently search for a solution space, promoting better models and eliminating weaker ones. The end process criterion may be defined in various ways. In our example, it is assumed that the evolution process may be stopped in two cases: if an acceptable model has been found or if the maximal number of populations is reached. An extensive evaluation of our approach on the real-life data extracted from the Bourse de Paris is published in [12].

4 Management of time series and generated experts

4.1 The time series database

The heart of the information management module is a database operated under the DBMS Oracle 8i. The database contains 13 tables organized in such a way as to minimize the response time. The conceptual schema of the database (Figure 2) shows the main tables and their relationships. To maintain a high quality of data all information to be inserted into the database undergoes a coherence test before writing them into the database. For instance, among these tests, one verifies if the high price is superior to all the other values of the day. The same operation is applied to the lowest value. If the data correspond well to all these constraints, they are stored in the database and are ready to be distributed to the users.

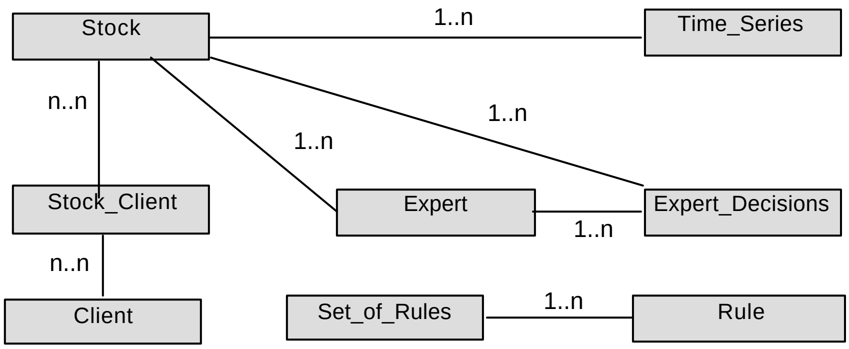

Figure 2. The database schema

Currently, the database contains the time series for all the principal stocks (forming the CAC40 index) which have been traded on the Paris Stock Exchange since January 2, 1997. Of course, a new time series may be easily inserted. The generated trading experts are also saved in the database in the table «Expert». The authorized users can download all these data from the database.

4.2 The system environment and software technologies

The latest software technologies have been used to implement the system [5,9,14]. The main idea was not only to keep the system maximally independent on the used computer platform and operating system, but also to offer an easy-to-use, friendly and intuitive user interface. It was necessary for the software to facilitate a distant connection to the database through the Internet and be easily modifiable. For these reasons, the following techniques have been used:

- the DBMS Oracle 8i was chosen to create a database system opened towards the e-commerce, Web or WAP applications,
- the language JAVA (SUN) was chosen to maintain the inter-platform software compatibility,
- the JDeveloper (Oracle) was used to develop Java programs; it has made it possible to integrate all necessary API to interoperate with the Oracle database,
- the library JDBC (Java Data Base Connectivity) has been used to assure the interconnection between the iBE and the database. The JDBC is a Java library for accessing DBMS. There are four types of JDBC drivers that enable a Java program to be connected directly or indirectly with a database. For our application, the driver Thin has been selected to work with the Oracle server or the execution platform of our application. This also makes possible to keep the same driver and the same application, whatever an execution platform may be.
- to develop a user interface, Sun proposes two libraries: the AWT (Abstract Windowing Toolkit, developed at the time of the first versions of Java), and the SWING library developed with the 1.2 version of Java. In our system, the last version of graphical interface is chosen.

4.3. Constraints of iBE execution

The iBE software, having been developed in Java language, requires installation of the Java Virtual Machine on a client station before running the application. Today a number of software editors have created their own virtual machines. In the project the Sun Virtual Machine is used. The software is available in the public domain via «Java Runtime Environment», version 1.30 for Windows, Linux and Solaris platforms.

5 User interface, simulations and predictions

In this section, the main user functions of iBE will be described. As mentioned in the preceding section, one of our major concerns was to realize an easy-to-use, friendly and intuitive interface. In fact, this feature is important for novice users that are not necessarily familiar with a computer environment.

5.1. Trading advice

Once the iBE system is correctly installed, the user may select his or her stock and visualize the stock and market behaviour. In Figure 3, the Alstom daily closing prices and the CAC40 index are displayed with information about signals generated by the Alstom trading expert (a red circle indicates a "Sell" signal, a green one points to a "Buy" signal). Below the diagram, iBE visualizes the daily quotes, the user decisions (columns with a yellow background), and the trading advice (columns with a blue background).

In the trading process, the user may act upon the expert advice, or he can make his own choice. The iBE system provides all information related to the user and expert decision, in particular the capital available each day, the number of stocks in hand and the total wealth. The user can perform more than one simulation at a time and save results for future analysis.

The generated experts have demonstrated a very good performance level on most of the financial time series. The global return on investment on 40 chosen stocks over on the testing set (Mai - July 2001) was in most cases better than the Buy-and-Hold strategy. On average, the percentage of correct decisions was about 65% but this measure has not been taken into account in the fitness function. More detailed evaluation of stock trading using genetic algorithms is given in [12].

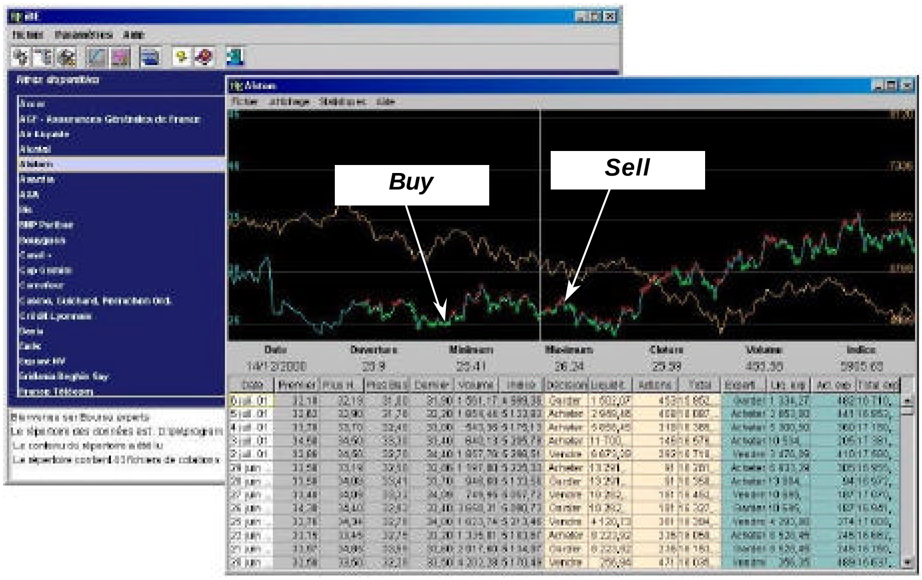

Figure 3. Alstom : Time series and trading decisions

5.2. Simulation of trading rules

The decision rules have been written into the knowledge base. In the simulation, the rule base contains 350 specific decision rules, generated from 80 generic rule templates. Of course, the user may use not only other structures of financial data but may also define his own set of decision rules. You can simulate any trading rule on the securities you choose. To provide easy-to-understand information about rule performance, the user may visualize the trading signals generated by a given rule over the whole period of a time series (Figure 4).

One of the interesting features of the system iBE is that the user may interactively modify the rule parameters and analyse the rule performance. Charts and rule decisions can be shown with ease - all with just one click.

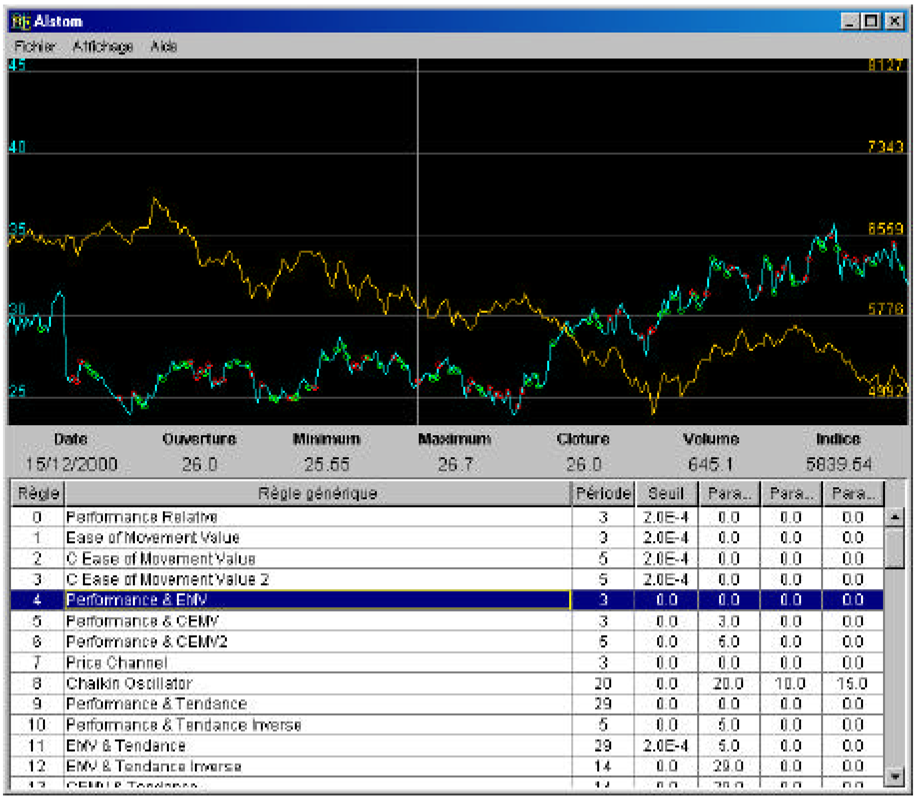

Figure 4. Trading decision generated by the rule « Performance and EMV »

5.3. The experts and their performances

The generated experts can be evaluated according to various criteria. Once a trading decision has been made, the user may evaluate a given expert over a defined period and compare the result with his trading strategy. The iBE system provides interactively essential information about the gain/risk factors, namely the return on investment, the daily return, the semi-variance, the standard deviation, the percentage of correct/incorrect decisions, the maximal loss, win/loss ratios, etc. (see Figure 5). This information is not only very useful in stock trading but also in estimating the quality of generated experts.

The genetic algorithm has a time-complexity linear with the length of the chromosome (the number of applied functions), the size of expert population, and the size of the time series. In the validation process a great number of genetically evolved models based on the different financial rules have been generated. The knowledge base, composed of over 350 decision rules concerning technical and economical situations, was used to generate approx. 250 000 experts for each stock to select the best one.

Computer time and cost were not critical issues. The learning and testing was performed on a Pentium III 1,2 GHz platform. Execution times to obtain satisfactory experts ranged from 4 to 6 minutes.

Figure 5. Expert evaluation

6. Conclusions

In this paper, the architecture and technology used to build the stock trading system have been presented. To generate trading decisions, the system applies genetically created trading experts. The powerful functions and easy-to-use interface help to take trading decisions, to manage financial time series efficiently, to simulate trading processes, to rank securities and find the ones that will work best with a given set of trading parameters. The approach has been validated on real-life data extracted from the Paris Stock Exchange database [12].

The genetic algorithm was able to prune effectively the very large search space, i.e. 2^{350} solutions for a 350-bits chromosome representation. Research results demonstrated that genetically evolved experts were able to trade much better than the gradient-based and the wavelet-based models. In contrast to gradient-descent algorithms, the genetic models are robust and are not trapped in local minimums. The genetic algorithm, in comparison with various statistical approaches, does not require restrictive assumptions such as stationarity, homogeneity and normal probability distribution of the trading variables concerned. The generated experts have simpler structures and are composed of relevant decision rules. Furthermore, in comparison to many other forecasting methods, the genetic algorithm is also capable of finding out the size of a sliding window, determining how far back in time the input sequence is correlated with the next prediction.

The database technology has assured efficient access to time series and generated expertise. Compared to the previous version of the system (http://www.bourse-experts.com), iBE, in addition, offers easy access to times series and trading experts, new graphic tools for trading simulation and rule evaluation, and a searching engine providing the latest financial news.

Currently, our research continues to investigate three areas. From the empirical point of view, our approach needs to be tested on other markets. More precisely, order driven or price driven markets may have different characteristics relative to technical trading. The second, more important orientation, is the application of genetic algorithms to intraday data. A trading prototype based on high frequency data using genetic algorithms will be available by the end of this year. The third area concerns portfolio management using evolutionary strategies: a paper is submitted for this conference.

References

[1] R.J. Bauer, *Genetic Algorithms and Investment Strategies,* Wiley, 1994.

[2] T. Bechu, E.Bertrand, *L'analyse technique. Pratique et méthodes.* Economica, 3me ed., 1998.

[3] P. Cahen, *Pratique de l'analyse technique*, Sefi, 1989.

[4] R. Caldwell, *Nonlinear Financial Forecasting*, Proc. of the First INFFC, Finance and Technology, 1997.

[5] G. Clavel, N. Mirouze, S. Munrot, *Java, La synthèse, Des concepts objets aux architectures Web*, Dunod, 2000

[6] W.Colby, T.Meyers, *The Encyclopedia of Technical Market Indicators*, Down Jones-Irwin, 1990.

[7] J. Creedy, L.M.Vance, *Nonlinear Economic Models*, Edward Elgar Publ., 1997.

[8] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[9] S. Feuerstein, B. Pribyl, C. Dawes, *Oracle PL/SQL Précis et concis*, O'REILLY, 2000

[10] J.O. Katz, D.L. McCormick, *Neurogenetics and its Use in Trading System Development*, NeuroVe$t Journ., Vol.2, No.4, pp. 5-11, 1994.

[11] J. Korczak, *Approche génétique pour découvrir un modèle de marché boursier*, in Actes de la Conf. d'Apprentissage CAP'99, pp. 9-16.

[12] J. Korczak, P. Roger, *Stock Timing using Genetic Algorithms*, Res. Rep. ULP, LSIIT, Illkirch, no 01/2000

[13] J. Murphy, *Technical Analysis of the Financial Markets*, NUIF, 1998.

[14] P.Y. Saumont, A. Mirecourt, *JAVA 2 Le guide du développeur*, Osman Eyrolles Multimedia, 2000

[15] A.S. Weigend, N.A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.