Evolutionary building of stock trading experts in real-time systems

Jerzy J. Korczak
Université Louis Pasteur
Strasbourg, France
Email: jjk@dpt-info.u-strasbg.fr

Piotr Lipinski
Université Louis Pasteur
Strasbourg, France
University of Wroclaw
Wroclaw, Poland
Email: lipinski@dpt-info.u-strasbg.fr

Abstract: This paper addresses the problem of constructing real-time stock trading expertise for financial time series. The expertise is arrived at by a genetic algorithm on the basis of a set of specified trading rules. As in most real-time expert systems, one of the main bottlenecks is the time constraint. In this paper two approaches were compared, the first based on 350 trading rules, and the second based on 150 particular linear combinations of these 350 rules. Experiments carried out on real data from the Paris Stock Exchange showed that focusing on only 150 rules highly reduced the computation time without significantly reducing the quality of the expertise.

1 Introduction

Evolutionary models are starting to appear and be used in stock trading [1, 2, 4, 7, 8, 9, 10, 14, 15]. In contrast to classical models, they do not assume a perfect market founded on the expected utility paradigm with homogeneous expectations of investors, normal rate of return distributions, no transaction costs, and a one-period investment horizon. The experimental findings have shown that most of these assumptions may not hold in reality. In our previous works, we have proposed an approach to discover a stock trading model from the financial time series based on evolutionary computing [7, 8, 9, 10]. In general, the evolutionary data mining techniques allow the creation of robust stock trading models not only from weekly or daily data but also from on-line quotations. In this paper, we will focus on the second case of trading decision: on-line trading advice.

The main idea of the evolutionary approach is to genetically create a stock trading model as true as possible to market dynamics, and to keep those dynamics updated and tuned to real-life situations as quickly as possible. In the model discovery process we assume that an investor generally maximizes expected utility; however, heterogeneous beliefs and some degree of irrational rules may be included in market simulations. The evaluation and fitness functions as well as the trading rules may be defined individually for each investor.

Evolutionary computations proved to be an efficient tool to solve a large number of complex optimization problems, which could not be done using classical methods due to the complexity of the problem or difficulty in defining a proper analytical model [5, 11]. They are also successfully used in finances. For instance, a genetic algorithm was applied to stock market prediction [9]. Evolution strategies were used for portfolio optimization [7]. A decision support system for real-time stock trading was designed on the basis of evolutionary intelligent agents [8, 10].

Real-time data analysis remains a grand challenge. Methods aspiring to be applied in real time should not only be efficient, but also quick to compute. Real-time intelligent experts should learn as quickly as possible, otherwise the efficiency of the methods may lose significance because the results will become delayed.

This paper is focused on elaboration of real-time stock trading expertise on the basis of a set of financial trading rules. Two rule sets are compared, the first consisting of 350 technical analysis rules, and the second consisting of 150 linear combinations of these 350 rules. Working with the smaller set leads to a significant reduction in computing time, but the question is whether or not it also causes a significant decline in the quality of the expertise.

This paper is structured in the following manner. It begins with the definition of stock market trading rules in Section 2. Section 3 describes data used in the algorithm. In Section 4 the problem of synthesizing expertise is defined. Section 5 focuses on evaluation of trading experts. In Section 6 the algorithm is briefly presented. Section 7 shows some experiments made on real data from the Paris Stock Exchange. Finally, Section 8 concludes this paper.

2 Stock Market Trading Rules

Technical analysis of financial data assumes that future values can be more or less accurately forecasted on the basis of past observations [3, 13]. Such analysis uses many functions to evaluate past data behavior and is able to detect trends and discover contexts leading to the occurrence of particular events such as the rise or fall of stock prices. There are a large number of trading rules based on technical analysis indicators [3, 13]. Using these rules, financial experts and market traders make decisions on the stock market: to buy, to sell, or to defer action and do nothing.

Technical analysis functions use a large amount of past data aggregated over a given period of time. Several rules are simple, requiring only a few recent quotations; thus, they are easy to formulate and not very time-consuming to compute. However, most rules work over longer periods and require more data, so computing them is more complex and time-consuming.

Formally, each trading rule is a function *f*, which computes a result $f(K_t) \in [-1, 1]$ on the basis of a knowledge K_t available at time *t*. The concept of a knowledge is rather abstract, but in a specific case the knowledge can be defined as a set of historical data, such as daily or intra-day stock quotations.

The result of a trading rule close to –1 corresponds to advice to sell, close to 1 to advice to buy, and otherwise indicates no recommendation, i.e. the advice is interpreted as to do nothing.

Formally, the result of a trading rule may be transformed into a trading decision $d_f(t)$ to sell, to do nothing, or to buy in the following way. Let

$$w(x) = \begin{cases} -1, & \text{if } x \le -\alpha_1 \\ 0, & \text{if } -\alpha_1 < x < \alpha_2 \\ 1, & \text{if } \alpha_2 \le x \end{cases}$$

for a given threshold $\alpha_1, \alpha_2 \in (0, 1]$. The decision of trading rule *f* at time *t* may be defined by $d_f(t) = w(f(K_t)) \in \{-1 \equiv sell,\ 0 \equiv do\ nothing,\ 1 \equiv buy\}$. Nevertheless, most trading rules directly return values of -1, 0 or 1, hence the further transformation of the result into a decision is not necessary.

The first example of a trading rule is the **Stochastic Oscillator** indicator introduced by G. Lane [3, 13].
Let

$$\kappa_t = \frac{Close_t - \min\{Min_t : t - T_1 < i \le t\}}{\max\{Max_t : t - T_1 < i \le t\} - \min\{Min_t : t - T_1 < i \le t\}}$$

$$\delta_t = \frac{1}{T_2} \sum_{i=t-T_2+1}^{t} \kappa_t$$

where $Close_t$, Min_t, Max_t respectively denote the close, minimal, and maximal price of the stock at time *t*. T_1 and T_2 are parameters. The selling signal is generated when κ_t rises above δ_t; the buying signal is generated when κ_t falls below δ_t.

The second example of a trading rule is the **Ease of Movement** indicator introduced by R.W. Arms [3, 13].
Let

$$R_t = \frac{Max_t + Min_t}{2} - \frac{Max_{t-1} + Min_{t-1}}{2}$$

$$V_t = \frac{Vol_t}{Max_t - Min_t}$$

$$\mathbf{EMV}_t = \frac{R_t}{V_t}$$

where Min_t, Max_t respectively denote the minimal and maximal price of the stock at time *t*, and Vol_t denotes the volume of transactions at time *t*. The selling signal is generated when ***EMV*** falls below a given threshold; the buying signal is generated when ***EMV*** rises above a given threshold.

Another example of a trading rule is the **Relative Strength Index** indicator introduced by W. Wilder [3, 13].
Let

$$\varpi_t = \frac{Close_t - Close_{t-1}}{Close_{t-1}}$$

$$U_t = \operatorname{avg}\{\varpi_i : t - T < i \le t \wedge \varpi > 0\}$$

$$D_t = \operatorname{avg}\{\varpi_i : t - T < i \le t \wedge \varpi < 0\}$$

$$\mathbf{RSI}_t = 1 - \frac{1}{1 + \frac{U_t}{D_t}}$$

where $Close_t$ denotes the closing price of the stock at time *t* and T is a parameter. The selling signal is generated when ***RSI*** falls below a given threshold; the buying signal is generated when ***RSI*** rises above a given threshold.

The number of well-known trading rules is large. In this paper, a set of 350 trading rules is considered. Moreover, new trading rules may be defined as combinations of other trading rules. For instance, one can introduce a new trading rule $f(K_t) = 0.41$ **EMV**$(K_t) + 0.59$ **RSI**(K_t) as a linear combination of trading rules **EMV** and **RSI**.

3 Data Description

Let *T* denote the length of the time period over which the calculations are performed.

Let ***K*** be a matrix of size *T* x *6* consisting of previous stock quotations. Columns correspond to opening, maximum, minimum, and closing prices, as well as volume and index values, respectively. Rows correspond to instants of the time period. That is, the *j*–th row constitutes the stock quotation at the *j*–th instant of the time period. Let $\boldsymbol{K_t}$ denote the matrix ***K*** truncated to the first *t* rows. It corresponds to the data available at the time *t*. The matrices ***K*** as well as $\boldsymbol{K_t}$ will be referred to as a knowledge.

Two sets of trading rules are considered in our experiments.

Let *N = 350* denote the size of the first set of trading rules $f_1, f_2, \ldots, f_N$ considered. Let ***X*** be a matrix of dimension *T* x *N* containing results of trading rules considered for each instant of the time period. The *i*-th column corresponds to the *i*-th rule, and the *j*-th row corresponds to the *j*-th instant of the time period. Let $\boldsymbol{X_t}$ denote the *t*-th row of the matrix ***X***. It corresponds to results of these trading rules computed on the basis of $\boldsymbol{K_t}$.

Let $M = 150$ denote the size of the second set of trading rules $g_1, g_2, \ldots, g_M$ considered. These rules are linear combinations of rules $f_1, f_2, \ldots, f_N$ defined by a matrix $\boldsymbol{Z}$ of size M x N containing linear coefficients of these combinations. In other words,

$$[g_1, g_2, \ldots, g_M]' = \boldsymbol{Z}\,[f_1, f_2, \ldots, f_N]'.$$

Let $\boldsymbol{Y}$ be a matrix of dimension T x M containing results of trading rules considered for each instant of the time period. The i-th column corresponds to the i-th rule, and the j-th row corresponds to the j-th instant of the time period. Naturally,

$$\boldsymbol{Y}' = \boldsymbol{Z}\,\boldsymbol{X}'.$$

Let $\boldsymbol{Y}_t$ denote the t-th row of the matrix $\boldsymbol{Y}$. It corresponds to results of these trading rules computed on the basis of $\boldsymbol{K}_t$.

The matrix $\boldsymbol{Z}$, which defines trading rules $g_1, g_2, \ldots, g_M$, comes from data preprocessing, which discovers dependencies within the set of trading rules $\{ f_1, f_2, \ldots, f_N \}$ and reduces the dimensionality of this set on the basis of general rule characterization studied over a long period of time using the principal component analysis (PCA) methods [6].

4 Problem Definition

In a given instant of time t, the trader bases his or her decision on certain trading rules. Depending on the choice of rules, the trader may get different advice. The question is which rules the trader should choose.

Some rules may be efficient during one time period and less efficient during some other time period. Some rules may work well only in combination with others. The trader would like to choose a set of rules according to criteria defined by his or her preferences, concerning for instance expected profit rate and risk aversion. The number of possible sets of rules is enormous, making the process of manual selection impractical.

Let e be a subset of the entire set of trading rules $\{ f_1, f_2, \ldots, f_N \}$. Such a subset will be referred to as a stock trading expert. In a natural way, the expert e can be presented as a binary vector of length N. The i-th coordinate of the vector corresponds to the i-th rule, where 0 stands for absence of the rule and 1 stands for presence of the rule.

A result $r_e(t)$ of expert e at time t is defined as the average of the results of trading rules included in the expert. Thus

$$r_e(t) = \frac{e'\mathbf{X}_t'}{e'\mathbf{1}}$$

where $\boldsymbol{1}$ stands for a vector of ones. The result $r_e(t)$ maybe transformed into a decision to sell, to do nothing, or to buy in a similar way that in the case of results of trading rules. Let $d_e(t)$ denote the decision of the expert e at the time t.

Naturally, all of the above considerations may be applied to the set of trading rules $\{ g_1, g_2, \ldots, g_M \}$ in an obvious way.

Thus, in a specific instant of time t, the trader looks for an expert. Before the trader chooses an expert, he or she assesses it. Experts may be evaluated and assessed according to given criteria. In order to formalize the assessment, a performance measure must be introduced.

Let $s_t(e)$ denote the performance of expert e at time t. Naturally, the function $s_t(e)$ depends on the time t at which the assessment is made. The performance measure can be defined in a variety of ways. Several examples are presented in the next section. The goal is to discover an expert e that maximizes the performance measure $s_t(e)$ at the given time t. The expert may be built by the genetic algorithm presented further.

This paper addresses the question of whether an expert generated on the basis of N rules $f_1, f_2, \ldots, f_N$ can compete with an expert generated on the basis of M rules $g_1, g_2, \ldots, g_M$, where $M \leq N$. Evaluation is made in terms of efficiency, whilst taking into consideration significant reductions in computing time.

5 Evaluation of Experts

Many performance measures were investigated. All of them were based on the expert's behavior in a specific time period. For instance, while the expert is being generated, its behavior over the last several time periods is assessed. After the expert has been generated, its behavior in the post-training period is assessed, based on data it did not see during its generation.

Assume that the behavior of the expert is evaluated in the period $[t_0, t_0 + h]$. Stock quotations this period are available at the moment of the evaluation.

A simulation is initially done. At time t_0 the expert receives an initial amount of money and an initial quantity of stocks, denoted by c_0 and s_0 respectively. The expert makes a decision $d_e(t_0)$, which is executed on the stock market at time $t_0 + 1$. If the decision is to sell, $q\%$ of possessed stocks is sold. If the decision is to buy, $q\%$ of possessed money is invested into stocks. The factor q is a parameter of the evaluation. Transactions on the stock market are usually charged some transaction costs, which the expert also pays. After the decision's execution, the expert's capital changes accordingly. At time $t_0 + 1$, the expert makes a decision $d_e(t_0 + 1)$, which is executed on the stock market at time $t_0 + 2$. Consequently, the expert's capital again changes. In this manner, a simulation of the expert's behavior over the entire period is obtained.

Performance measures refer to the expert's capital in the period considered. They not only draw attention to the profit, but also the risk connected with its making. Moreover, they study the profit taking into consideration

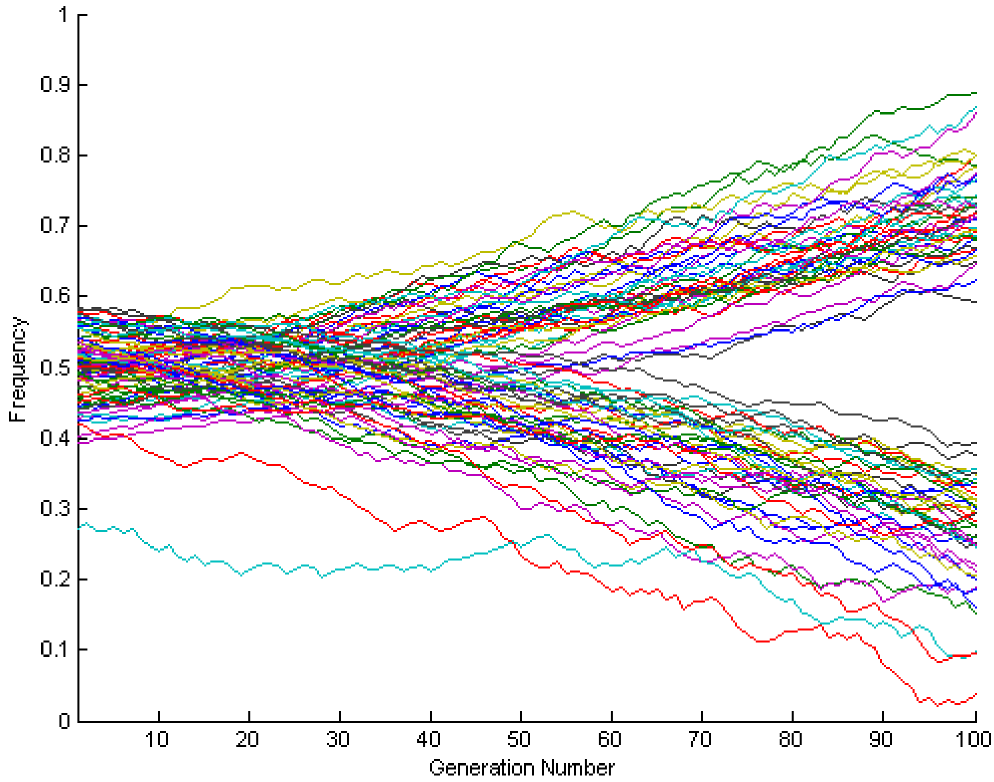

Figure 1. Frequency of rules occurring in 100 populations of experts

the market condition defined by various indices (e.g. the market index).

The first example of a performance measure is the Sharpe's ratio [4, 12],

$$s_t(e) = \frac{\mathbf{E}(R - I)}{\sqrt{\mathbf{Var}(\mathrm{R} - I)}}$$

where R denotes the return rate of the expert and I denotes the rate of the market index at time t.

Another example of a performance measure is the Sterling's ratio [4, 12],

$$s_t(e) = \frac{\mathbf{E}(R)}{d}$$

where R denotes the return rate of the expert and d denotes the drawdown of the expert at time t.

More performance measures can be found in [4, 12].

6 Algorithm

The algorithm was constructed on the basis of a genetic algorithm [5, 11]. It is described in detail in [9]. Due to size constraints, only the highlights are presented here.

6.1 Population

The population consists of experts represented by binary strings. It may be initialized randomly or on the basis of experts generated at the previous instant of the time period.

6.2 Evaluation

Many objective and fitness functions were investigated. The evaluation of an expert is performed according to the procedure described in the previous section.

Let t denote the instant when the expert is being generated. Behavior of the expert is studied in the period $[t - h, t - 1]$, for a given h (e.g. $h = 60$). Objective functions are defined by chosen performance measures (e.g. the Sharpe's ratio).

6.3 Termination Conditions

Termination conditions include a constraint on the number of iterations as well as homogeneity conditions.

6.4 Convergence of the Algorithm

It is interesting to watch the evolution of experts to observe how often a given rule occurs in them. The frequency of rules occurring in expert populations is presented in the Figure 1. One can see that the algorithm separates efficient rules from inefficient ones. Inefficient rules die out in the evolution.

7 Experiments

The goal is to compare two approaches, the first dealing with generating experts on the basis of N trading rules f_1, f_2, …, f_N, and the second dealing with generating experts on the basis of M trading rules g_1, g_2, …, g_M. Trading rules g_1, g_2, …, g_M are linear combinations of rules f_1, f_2, …, f_N defined during data preprocessing on the basis of general rule characterization.

Experiments were performed on four financial time series from the Paris Stock Exchange. Each financial time series includes real-time quotes of a given stock aggregated over a period of one minute. Table 1 presents the data considered.

Table 1. Financial Time Series

Stock	Time Period
AXA	Jan 1, 1998 – May 12, 2003
Credit Lyonnaise	Jan 1, 1998 – May 12, 2003
Peugeot	Jan 1, 1998 – May 12, 2003
STMicroelectronics	Jan 1, 1998 – May 12, 2003

Each experiment concerned one of the stocks presented in Table 1. It began with randomly choosing an instant of time t from the time period specified in the second column. Afterwards, two experts were built for time t (on the basis of the knowledge K_t available at time t). The first expert, e_1, was built according to the first approach, while e_2 was built according to the second approach. The time period, as well as the parameters of the algorithm, was the same in both cases. The optimization was performed with respect to the same objective function. Next, the behavior of each expert was studied on a specific time period – the training period $[t - h, t - 1]$ for a given h (e.g. $h = 60$) or the post-training period $[t, t + h]$ for a given h (e.g. $h = 5$). The results obtained were used to compare the efficiency of experts. Besides the efficiency, the computing time necessary to generate the expert was analyzed.

Table 2 presents a summary of the first part of the experiments, showing the expert evaluation during the training period $[t - h, t - 1]$, where $h = 60$. This kind of evaluation is normally used in the genetic algorithm presented to evaluate the population of experts.

The second part of the experiments is summarized in Table 3. These experiments evaluated the expert during the post-training period $[t, t + h]$, where $h = 5$. They assessed performance of the expert on stock market data that had not been used during the training process.

Table 4 shows the computing time necessary to generate the two experts in the experiments presented above.

Table 2. Performance during training periods

Stock	Objective function *s*	*t*	*s(e1)*	*s(e2)*
Peugeot	Sterling's ratio	February 1, 2003 10:31	0.34	0.33
Credit Lyonnaise	Sharpe's ratio	December 17, 2001 14:28	0.87	0.84
AXA	Sterling's ratio	November 15, 2002 9:43	0.42	0.36
STMicroelectronics	Sharpe's ratio	October 5, 2002 16:17	0.28	0.28
Peugeot	Sharpe's ratio	May 10, 2002 14:56	0.63	0.64
AXA	Sterling's ratio	June 11, 2002 12:09	0.91	0.88

Table 3. Performance during post-training periods

Stock	Objective function *s*	*t*	*s(e1)*	*s(e2)*
Peugeot	Sterling's ratio	February 1, 2003 10:31	0.12	0.11
Credit Lyonnaise	Sharpe's ratio	December 17, 2001 14:28	0.43	0.39
AXA	Sterling's ratio	November 15, 2002 9:43	0.38	0.36
STMicroelectronics	Sharpe's ratio	October 5, 2002 16:17	0.09	0.12
Peugeot	Sharpe's ratio	May 10, 2002 14:56	0.59	0.58
AXA	Sterling's ratio	June 11, 2002 12:09	0.39	0.31

Table 4. Computing time

Stock	Objective function *s*	*t*	*time(e1)*	*time(e2)*
Peugeot	Sterling's ratio	February 1, 2003 10:31	19 s	11 s
Credit Lyonnaise	Sharpe's ratio	December 17, 2001 14:28	21 s	12 s
AXA	Sterling's ratio	November 15, 2002 9:43	18 s	9 s
STMicroelectronics	Sharpe's ratio	October 5, 2002 16:17	26 s	14 s
Peugeot	Sharpe's ratio	May 10, 2002 14:56	23 s	11 s
AXA	Sterling's ratio	June 11, 2002 12:09	17 s	9 s

8 Conclusions

This paper presents a study of a genetic algorithm that builds stock trading experts on the basis of technical analysis trading rules. Two approaches were compared. The first concerned building experts on the basis of 350 trading rules, while the second concerned building experts on the basis of 150 other trading rules that were linear combinations of these 350. Focusing on only 150 rules highly reduces the computation time, which is an important aspect of all real-time systems. The question to be answered was whether replacing the original set of rules by a set of linear combinations of them causes a significant decline in performance of the experts.

Experiments performed on real data from the Paris Stock Exchange showed that the original set of rules can successfully be replaced with the smaller set of linear combinations of them without a major efficiency loss, while achieving a significant reduction in computing time. This observation enables optimization of the generation of stock trading experts in real-time systems, such as presented in [8, 10], as the computing time is always one of the main constraints in real-time data processing.

References

[1] Allen, F., Karjalainen, R., "Using Genetic Algorithms to Find Technical Trading Rules", Journal of Financial Economics, 51, 1999, pp.245-279.

[2] Bauer, R., J., "Genetic Algorithms and Investment Strategies", Wiley, 1994.

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

[4] Dempster, M. A. H., Jones, C. M., "A Real-Time Adaptive Trading System using Genetic Programming", Quantitative Finance, Vol. 1, 2001, pp. 397-413.

[5] Goldberg, D. E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989.

[6] Jolliffe, I., T., "Principal Component Analysis", Springer, 1986.

[7] Korczak, J., Lipinski, P., Roger, P., "Evolution Strategy in Portfolio Optimization", Artificial Evolution, ed. P. Collet, Lecture Notes in Computer Science, vol. 2310, Springer, 2002, pp. 156-167.

[8] Korczak, J., Lipinski, P., "Shortening of Evolutionary Agent Learning using Sequential Pattern Methods", Proceedings of the AISB'03 Third Symposium on Adaptive Agents and Multi-Agent Systems, AAMAS, ed. D. Kazakov, D. Kudenko, E. Alonso, University of Wales, Aberystwyth, UK, 2003, pp.115-119.

[9] Korczak, J., Roger, P., "Stock timing using genetic algorithms", Applied Stochastic Models in Business and Industry, 2002, pp. 121-134.

[10] Lipinski, P., Korczak, J., "Design of Stock Trading System based on Intelligent Agents", Proceedings of 9th International Multi-Conference on Advanced Computer Systems, Production System Design, Supply Chain Management and Logistics, ed. A. Dolgui, J. Soldek, O. Zaikin, Technical University of Szczecin, Szczecin, Poland, 2002, pp.177-186.

[11] Michalewicz, Z., "Genetic Algorithms + Data Structures = Evolution Programs", Springer Verlag, New York, 1994.

[12] Moody, J., "Performance Function and Reinforcement Learning for Trading Systems and Portfolios", Journal of Forecasting, 1997.

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

[14] Neely, C., Weller, P., Dittmar, R., "Technical Analysis in the Foreign Exchange market Profitable? A Genetic Programming Approach", Journal of Financial Quantitative Analysis, 32, 1997, pp.405-426.

[15] Tsang, E. P. K., Li, J., Markose, S., Er, H., Salhi, A., Iori, G., "EDDIE In Financial Decision Making", Journal of Management and Economics, Vol.4, No.4, 2000.