ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2005, Vol. 34, No. 3

COMPUTATION IMPROVEMENT OF STOCK MARKET DECISION MAKING MODEL THROUGH THE APPLICATION OF GRID

Jovita Nenortaitė

Informatics Department, Vilnius University Kaunas Faculty of Humanities
Muitinės 8, 44280 Kaunas, Lithuania

Abstract. The paper is focused on the presentation of decision - making model for stock markets. As well paper introduces grid computation idea and discusses the possibilities of grid technology application for the optimization of computational power related to the realization of the proposed model. The introduced decision - making model is based on the application of artificial intelligence tools: artificial neural networks (ANN) and particle swarm optimization algorithm (PSO). While making computations with the proposed decision - making model the decision computation process takes quite a lot of time. Working with stock markets it is very important to make decisions fast. One of the possibilities to do that is to apply grid technologies for the distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations.

Key words: Stock Markets, Artificial Intelligence, Swarm Intelligence.

1. Introduction

The continuing improvements of computer technologies make a big influence on the globalization of stock markets. The complexity and "noisiness" of stock markets cause difficulties in making real time analysis of it. For decision making in stock markets there is not enough to use conventional tools and more efficient its information processing tools are required. That means that new computerized decision making tools have to be used in order to create powerful decision–making and prediction systems. The use of artificial intelligence had made a big influence on the forecasting and investment decision–making technologies and it was proved that some efficient results can be obtained [8], [9]. As it is claimed by [4] in the comparison with conventional statistic tools the main advantages of artificial intelligence tools are that they are able to learn to recognize patterns in the data set, they are flexible in a changing environments and they can build models when more conventional approaches fail. The advantages of Artificial Intelligence tools applications are as well presented by Mirmirani and Li [10]. They are claiming that in comparison with statistical techniques, ANN makes less restrictive assumptions on the underlying distribution. Another artificial intelligence tool, which already have shown promising results of its applications, is Swarm Intelligence (SI). The examples of SI applications were presented by Pavlidis, Tasoulis and Vrahatis [13], Carlisle and Dozier [3], Po-Chang and Ping-Chen [12]. It was proved that having complex systems a collection of individuals often solve a problem better than an individual - even an expert [1], [6]. SI includes two algorithms: Ant Colony Optimization (ACO) algorithm and PSO algorithm. ACO algorithm is used for solving computational problems which can be reduced to finding good paths through graphs. Mainly ACO algorithms have been used to produce good near–solutions to the traveling salesman problem. The biggest number of this algorithm applications appear in the field of telecommunications, network routing etc. For the decision–making in stock markets it is more suitable PSO algorithm as it allows to make the search of "best" ANN on a current time and make decisions, based on its performance. The application of ANN is quite common in financial markets. The application of this artificial intelligence tool is believed to be an effective modelling procedure when the mapping from the input to output vector space contains information about the modeled object. However, despite the optimistic view concerning the possibilities of ANN in financial forecasting [4], [8], [9], our investigations have showed that the average prediction quality was not efficient using feed-forward ANN and also ANN with pseudo–input variables (using principal component) [2]. The introduced decision–making model combines ANN and PSO algorithm application. ANN are used to make the analysis of daily stock returns and to form one step forward decision for the purchase of stocks. While PSO algorithm is applied for the training of ANN. The PSO algorithm was chosen to use as PSO is the only evolutionary algorithm that does not incorporate survival of the fittest, which features the

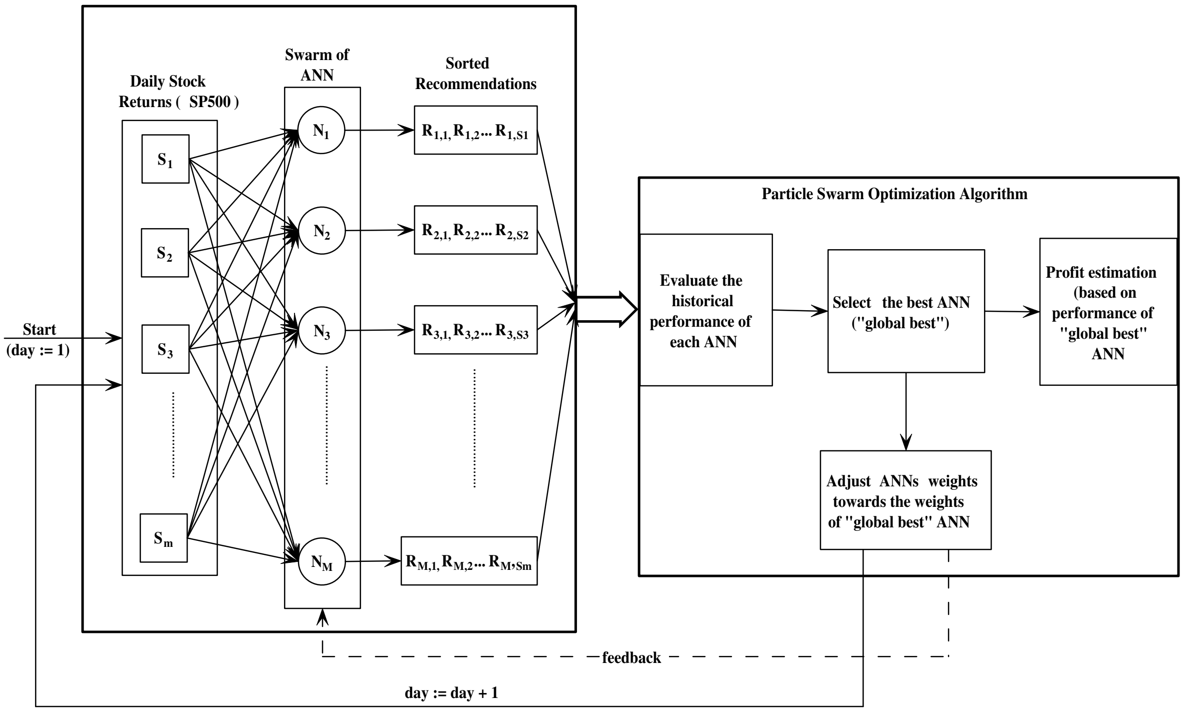

Figure 1. Scenario of Decision Making Model

removal of some candidate population member [5]. In the proposed decision–making model this feature of PSO learning process is very important. It is so as in some cases working with quickly changing (so called "noisy") environments the "bad" particles (particle with lower fitness) can become "good" once and give the right decision. The experimental investigations have shown that a PSO algorithm is better for applications that require fast learning algorithms. As well it was proved that PSO algorithm requires less number of computations to achieve the same error goal as with that backproagation [7]. The objective of this paper is to introduce a decision-making model for one day forward decision making in stock markets. The paper presents the structure of decision–making model, which is based on the application of ANN and PSO algorithm. Another objective of the paper is to introduce grid computation as it could be one of the solutions, how to increase the computation speed while making decisions. The paper is focused on the presentation of the general architecture of grid and discussion on the possibilities of grid computation technology application for the computations of decision - making model.

2. Decision Making Model

The decision–making model combines the application of ANN and PSO algorithm. The layout of decision–making model is presented in Figure 1. For the model realization there are used historical data of some group of stocks daily returns for defined period of time. The user of decision–making model can define its own group of stocks and time period. It is preferable that selected stocks would have a high liquidity ratio, which is related with low commission rates while making stock trading.

The decision–making model combines the application of ANN and PSO algorithm. The layout of decision–making model is presented in Figure 1. For the model realization there is used a historical data of some group of stocks daily returns for defined period of time. The user of decision–making model can define its own group of stocks and time period. It is preferable, that selected stocks would have a high liquidity ratio, which is related with low commission rates while making stock trading.

The historical data of daily stock returns are passed to the swarm ANN. In the proposed decision making model there are used single layer ANN (see Figure 2). It was decided to use such ANN in order to check the suitableness of the proposed decision–making model in general. After the data are passed to ANN the net result is a linear combination of each of the weighted input vectors:

$$x = \sum w_j k_j \qquad (1)$$

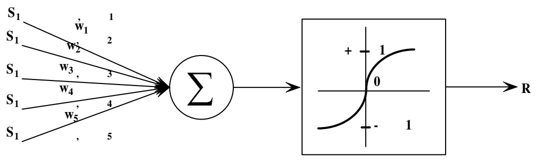

Figure 2. Single Layer ANN

Here w_j represents the weights, and k_j - the daily stock returns. The weights of ANN are initialized randomly at the beginning of the procedure:

$$w = rand(w, n) - 0.5 \tag{2}$$

w are randomly initialized weights of ANN and n is the number of ANN. The start weights are kept relatively small, and their mean is close to zero. In the decision–making model there are considered stock returns for $k = 1 : 5$ interval of days. Such a time period was chosen as it has the biggest influence on the next day stock returns [13]. For example, if stock returns of 5 days are negative, there is a probability that the returns of these stocks will increase on the next day and vise versa. The detailed description of k step selection is presented in the section of experimental investigations. Further, for each day and each stock the buy/sell recommendations are calculated:

$$R = \frac{2}{1 + e^{-x}} - 1 \tag{3}$$

The recommendations R represent the relative rank of investment attraction to each stock in the interval [-1, 1]. The values -1, 0 and 1 represent recommendations: sell, hold and buy, respectively. After, all the stocks are sorted according to the calculated recommendations in the descending order. Having sorted stocks, we are taking into account the first 3 stocks that have the highest recommendations. As the next step, the observation of these stocks on the next day follows. The returns of these stocks are included into the sliding window profit estimation expression. The idea of sliding window is presented in Figure 3. It is important to mention that the data, which were used for the calculation of recommendations it is not further used for the training of ANN. As it can be seen from the Figure 3, the sliding window is divided into two parts: training and decision making. In the training part there are selected the ANN with the best performance (the highest total profit for the selected sliding window) and the "worst" ANN are trained towards the behavior or these selected "global best" ANN. The training part is based on the application of PSO algorithm, which is presented in Figure 1. The final decision of "buy", "sell" or "hold" is made according to

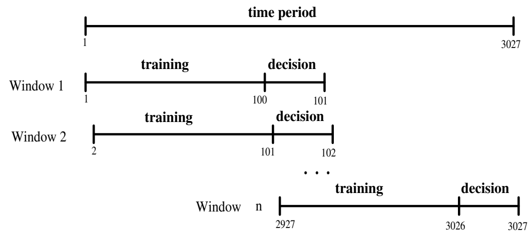

Figure 3. Sliding Window

the recommendation of "global best" ANN. Each day the selection of “global best” ANN is made comparing the performance (fitness function) of all ANN and selecting the network with the highest performance. This network is used for further experimental investigations (for the calculation of returns on the next day). When the “global best” ANN is known the training of other ANN is made. The ANN are trained by adapting the ANN weights towards the weights of "global best" ANN. The ANN weights are recalculated according to the following equation:

$$w_{new} = w_{old} + c * (w_{best} - w_{old}) \tag{4}$$

where w_{old} presents the weight of the explored ANN, w_{best} is the weight of “global best” ANN and c - learning rate. The choice of learning rate is crucial as the exploring ability of PSO may cease to exist. The training of ANN lets to move towards the best solution. It is important to mention that for each day the different “global best” ANN may be selected. The experimental investigations are discussed in Section 3.

3. Decision - Making Model Evaluation

All the experimental investigation were run according to the above presented scenario (see Figure 1) and were focused on the estimation of possible returns, which could be got while applying our proposed decision - making model. In all the cases the expected returns were calculated considering transaction fees that are unavoidable in the stock markets. The analysis of the commission fees of different e-brokers showed that the commission fee in real trading process is usually between 0.15 % - 0.3 %. For example, such transition fees are provided by the company of Interactive Brokers. Having bigger selling and buying volumes this fee could be even smaller - 0.1 %. Based on that and our made experimental investigations for the further investigations we are considering the commission fee which is equal to 0.15 %. We are making an assumption that each day we are paying 0.15 % of commission fee for buying new stocks. The value got on the last investigated trading day is considered as the profit. The profit is calculated as a sum of stock returns (%). In all experimental investigation the training of ANNs was made through the adjustment of ANNs weights towards the weights of “global best” ANN. For the adjustment of the weights there was chosen the learning rate of 0.05. Such value was selected based on the experimental results which are presented in the article [11]. As well it is important to mention, that for all experimental investigation, the sliding window of 100 days was used. The results of experimental investigations are presented in Figure 4. In Figure 4 there

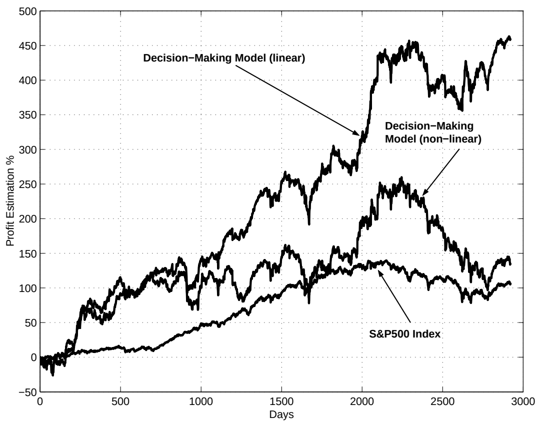

Figure 4. The Results of Model Evaluation

is presented the comparison of the results while having three cases: conservative investment approach (investments are done into SP500 index by buying and holding it for all time period), and decision making model (linear case) application and decision making model (non-linear case) application. In the case of non-linear model application there are used multi layer ANN. The presented results show that the application of PSO algorithm and training of ANNs towards the performance of the “global best” ANN give quite good results. Compare to the conservative approach the proposed decision making model lets to achieve almost 5 times better results (having the linear case of the model). The reason of that is that every day the investment decisions are made using the “global best” particle and at the same time all other particles are slightly moved towards it. Such training of ANNs ensures the movement towards the best decision. In the case of non-linear decision making model the results are not so good. That lets us to make an assumption that in complex systems the more advanced decision making model not necessarily give better results. Non-linear case of decision making model is too complex and its different realizations can have big variations.

However, while making computations with the proposed model we have faced some problems. The big amount of data and complex computations do not allow us to ensure the fast decision making which is necessary while working in stock markets. There can be mentioned the following problems:

1. Computational power. In order to achieve reasonable results, the computation process has to be fast. The big amount of data and complex computation algorithm does not allow us to ensure this feature of the proposed decision - making model.

2. Possibility to run several realizations of the model at the same time. In order to ensure good quality of the proposed decision - making model we have to make experimental investigations almost on each variable application case. There is no possibility to run several realizations of the model at the same time in our case.

3. Accessibility of the data. Currently, there is a group of researches, working on the same topic: stock market analysis, forecast and decision making. Most of the researchers are using the same data for the experimental investigations. The possibility to access the same data set in the same location could ensure the more powerful computational process and could release researches from the unnecessary actions, like data set formation, copying etc. As well it could ensure the more adequate evaluation of different models.

In order to eliminate the above mentioned problems, we were looking for possible solutions. As one of the solutions for this problem we found out a grid technology application. Further, the paper deals with a description of grid technologies in general, and its application for the optimization of the proposed model computations.

4. Grid Architecture

Grid computing is a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations. Grid technology is widely used for solving complex computational problems. Grid computing is an evolving area of computing and its standards and technologies are still under the development procedure. Architecturally grid can be seen as a group of interconnected clusters (computing resources) and data storage nodes. The computing nodes and data storage nodes are connected via GRID middleware. In Figure 5 there is presented the scenario of grid technology which is based on the the "NorduGrid" applica-

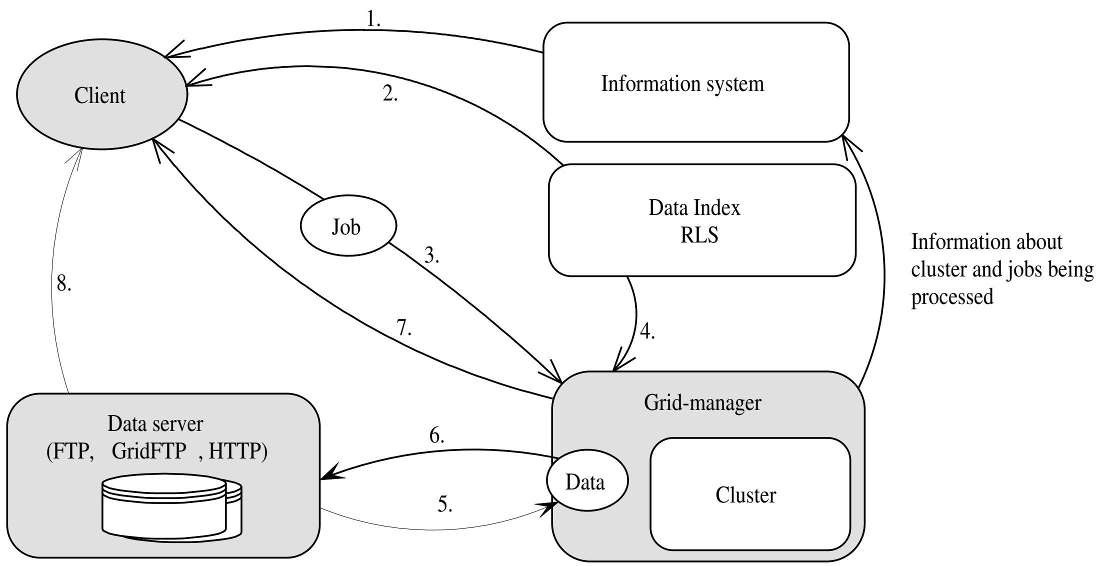

Figure 5. Grid Technology Application

tion case [15]. The presented scheme illustrates the interaction processes between grid and client. As well it describes the processes between different parts of the grid. All the grid computation process can be divided into several main steps. In the first step, knowing the job the client has to gather the information about possible resources and data storage locations. The processes of information gathering are presented by the first two arrows (1 and 2). When this information is known the second step of computation is started. In the second step the client selects the suitable computing resources and creates job description. The formulated job is sent to the cluster (arrow 3). In the cluster there are gathered the data of the job. When the job and data are known the third step starts and the computation process is executed (4 and 5 arrows). In the last computation step, after the execution of the job, the data are sent to a data server (arrow 6). After the client detects that the job is completed, he/she retrieves the result data from the data storage (arrow 8) and removes the job from the cluster. Usually, if the job from the cluster is not removed by the client, it is removed automatically after some period of time. As it can be seen from Figure 5 there are the following grid components:

1. Data index RLS (Resource Location Server) - stores information about available data storage nodes, available data and data locations.

2. Information system - it is an information system server that maintains and provides the information about the installed clusters and their computing resources. For each cluster it maintains information about the installed hardware, available software. Information system servers also maintain the status information. The status information includes information about currently running jobs and current usability of the cluster.

3. Job - is users described task. Job description and description language differs from middleware to middleware. Generally, job description defines the required software and data, which is needed to complete the task. Additionally, it provides requirements for a task to be completed. The provided information may include maximum time required to complete the task, required hardware, processor power, space on a data storage node. This information is used by Grid middleware to locate the most appropriate clusters and data storage nodes. If at the current time the required resources are not available, then the submitted job is added to a queue. Only when all required resources will be available the job will be started. Also the time when the job will be started depends on job management policy used in grid middleware.

4. Data server - is the data storage node. In the data server there are stored data, which are needed for performing the jobs of the client. Data are treated as a resource. Some data can be stored in one location and be used by several clients for their submitted jobs.

5. ARC (Advanced Resource Connector) Grid-manager - manages resources, assigns resources to the submitted job, controls job execution queue. After all the required for a job resources are available, the job with description of all the resources is passed to a cluster and executed.

6. Client - Clients usually are lightweight clients. The main task for a client is to provide means to submit jobs, view the status of a job and retrieve results after the job is finished.

5. Computation Using Grid

It is important to understand the origin of grid computing in order to take the full advantage of the grid. All the jobs submitted to a grid are finally executed in the clusters. In the upper layer the grid is seen as a set of computational nodes (clusters) and data storage nodes. The cluster by itself is a collection of computers hooked together by a network. In this way each cluster is a large parallel computer.

There can be separated two cases of the application realization on a grid. First, the application could be modified so that it could take an advantage of the grid. It is good when the application task can be divided into loosely coupled subtasks [15]. In that case each subtask can be executed in different clusters. As well each subtask could be a concurrent program by itself. Concurrent program would take a full advantage of computational power provided by a cluster. But it is important to know that not all applications can be easily parallelized. Currently, in order to solve this problem there are developed automated tools for the parallelization of the task. These tools are still under development, but in future they should help to parallelize the task using different parallelization algorithms. As it was mentioned before, our introduced decision - making model is based on the application of ANN and PSO. The computations are made taking into account many variables which are closely related to each other. Taking into account the above mentioned aspect of dividing tasks into the sub - tasks we have faced the problem of dividing decision - making algorithm into several smaller algorithms. The only possibility is to divide the algorithm so, that the deltas of daily stock returns could be calculated on one cluster and afterward sent for further calculations to other clusters.

In the other case of running the application on the grid no changes to the application have to be made. As the dividing procedure of our proposed model algorithm is too complicated and even not possible, this feature of the grid lets us avoid the segmentation of the application. Here we can take an advantage of the grid, having the possibility to run several realizations of the model on the different clusters. This possibility is very important to us as we can ensure the more effective analysis of the model. In order to ensure good quality of the proposed decision - making model we have to make experimental investigations almost on each variable application case. Another advantage of the grid application could be the data storage possibility. This feature could be very useful if there is a group of researches working with the same data set.

On a cluster bias usually there is installed some kind of message passing environment. The most common environments are the implementations of Message Passing Interface (MPI) specification or PVM (Parallel Virtual Machine) software systems [16]. These systems provide communication means for interconnected computers and in this way the tasks of one application can be distributed among other computers. Other applications like mathematical packages or other computational tools are built on top of PVM or MPI.

6. Conclusions

Having complex computations and working with a big amount of data we are forced to look for the possibilities how to increase the computational power of the proposed applications. In the paper there was introduced a decision - making model, which is based on the application of ANN and PSO algorithm. While making an investigation of the proposed model we have faced the problem of computational power. As one of the possible solutions to this problem we have found the use of a grid. The paper had introduces the grid architecture and there were discussed the possibilities of making the decision - making model computations on it. The first investigations on the grid application possibilities for the computation improvement of decision - making model have showed that it is possible to reorganize the application so that it could be run on a grid and the computation speed of the model could be improved. In future we intend to try to make decision - making model computations on a grid.

REFERENCES

[1] **K. Bartholdson, J.M. Mauboussin.** Thoughts on Organizing for Investing Success. *Credit Suisse First Boston Equity Research*, 2002.

[2] **E. Blandis, R. Simutis.** Using Principal Component Analysis And Neural Network for Forecasting of Stock Market Index. *Bizinesa augstskola Turiba SIA, Riga*, 2002, 31-35.

[3] **A. Carlisle, G. Dozier.** Adapting Particle Swarm Optimization to Dynamic Environments, *ICAI Conf. on Artificial Intelligence*, Las Vegas, USA, 2000, 199-204.

[4] **A. Din.** Optimization and Forecasting with Financial Time Series, *Note from Seminar at CERN*, 2002.

[5] **R.C. Eberhart, Yuhui Shi.** Comparison Between Genetic Algorithms and Particle Swarm Optimization, *Evolutionary Programming*, 1998, 611-616.

[6] **N.R. Franks.** Army of Ants: A Collective Intelligence, *American Scientist*, 77(2), 1989, 138-145.

[7] **V. G. Gudise, G. K. Venayagamoorthy** Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks, *Proceedings of the 2003 IEEE on Swarm Intelligence Symposium*, 24-26 April, 2003, 110 - 117.

[8] **T. Hellstrom** Optimizing the Sharpe Ration for a Rank Based Trading System, *Lecture Notes in Artificial Intelligence*, LNA 2258 Springer-Verlag, New York, 2001, 142-155.

[9] **I. Kaastra, B. Milton** Designing a Neural Network for Forecasting Financial and Economic Time Series, *Neurocomputing*, 10(3), 1996, 215-236.

[10] **S. Mirmirani, C.H. Li** Gold Price, Neural Networks and Genetic Algorithms, *Computational Economics*, 23(2), 2004, 193-200.

[11] **J. Nenortaite, R. Simutis.** Stocks' Trading System Based on the Particle Swarm Optimization Algorithm, *Lecture Notes in COmputer Sciene, Vol. 3039*, 2004, 843-850.

[12] **K. Po-Chang, L. Ping-Chen.** A Hybrid Swarm Intelligence Based Mechanism for Earning Forecast, *Proceedings of the 2nd International Conference on Information Technology for Application*, 2004, 193-198.

[13] **N.G. Pavlidis, D. Tasoulis, M.N. Vrahatis.** Financial Forecasting Through Unsupervised Clustering and Evolutionary Trained Neural Networks, *2003 Congress on Evolutionary Computation*, Canberra Australia, 2003.

[14] **R. Simutis.** Stock Trading Systems Based on Stock's Price Ranks (in Lithuanian), *Ekonomika*, 62, 2003, 157-164.

[15] **"NorduGrid" homepage.** http://www.nordugrid.org/papers.html

[16] **Local Area Multicomputer/Message Passing Interface (LAM/MPI)**. http://www.lam-mpi.org