HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

A. Rubchinsky, D. Chubarova

EFFECTIVE STOCK MARKET TRADING ALGORITHM: A RETROSPECTIVE ANALYSIS BASED ON S&P-500 DATA

Working Paper WP7/2025/01
Series WP7

Mathematical methods
for decision making in economics,
business and politics

Moscow

2025

Rubchinsky A., Chubarova D.
Effective Stock Market Trading Algorithm: a retrospective analysis based on S&P-500 data [Electronic resource]: Working Paper WP7/2025/01 / A. Rubchinsky, D. Chubarova; HSE University. – Electronic text data (900 Kb). – Moscow: HSE Publishing House, 2025. – (Series WP7 “Mathematical methods for decision making in economics, business and politics”). – 37 p.

The article examines one of the most famous examples of socio-economic systems, characterized by significant uncertainty – the S&P-500 stock market, where shares of 500 largest US companies are traded. No assumptions are made about the probabilistic characteristics of the stock market. A flexible algorithm for daily trading has been developed, based on both known fixed data on the value of shares in previous days and on income and expenses already received since the beginning of the current quarter. The numerical experimental results demonstrate a steady increase in accumulated income in 1995-2014, as well as the possibility of using pre-crisis and post-crisis periods to reliably increase income from stocks trading.

A. Rubchinsky, D. Chubarova (HSE University)
arubchinsky@yahoo.com;
dchubarova@hse.ru

1. Introduction

In numerous articles (see, for example, [1-6]) devoted to stock market trading algorithms, one basic assumption is made (explicitly or implycitly): **stock price changes are described by some stochastic process.** The researchers' efforts are focused on identifying and analyzing these processses. At the same time, earnings from trading stocks selected on the basis of this basic assumption are also considered as the realization of stochastic processes.

Reviewing publications in this field is not the purpose of the article. It can be noted that the wide variety of proposed methods, problem statements, and algorithms most likely indicates a certain dissatisfaction with both existing probabilistic models and the financial results obtained on their basis. Furthermore, the processes in the stock market can be considered as probabilistic with a great reserve. At the same time, it is safe to speak about the uncertain nature of these processes. It is precisely this uncertainty that makes it necessary to develop new approaches that do not rely on assumptions about the probabilistic nature of the functioning of stock markets.

In this paper, we propose a new algorithm for trading on the stock market that does not use any probabilistic assumptions about previous stock prices. All of them are considered simply as fixed numerical data, entirely known at the end of the next working day. However, the algorithm for determining stocks for trading on the next business day uses a standard random number generator, so that the daily income received is a "legitimate" random variable. And if in one day we can talk about mathematical expectation and other traditional probabilistic parameters, then when determining the accumulated values of this random variable for a quarter, a year, and over long periods of time, these amounts demonstrate very good convergence under different implementations of the specified correct random process. Since these amounts take on noticeable positive values (based on a small number of traded shares), we can talk not only about the stability of the proposed trading algorithm, but also about its effectiveness.

The work is structured as follows. Section 2 provides a brief description of the S&P-500 stock market and provides a general flowchart of the quarterly trading algorithm (QTA) over a fixed period of time (in this case, one quarter). The main algorithms included in the QTA are:

A. A new clustering algorithm that does not build the only "most correct" graph partition into subgraphs, but offers a family of such partitions from which you can choose one using a variety of considerations. In particular, the method does not reject partitions with a very large ratio of the powers of the parts, when this is an objective property of the systems under consideration. The algorithm is applied to a well-known market graph [7], which is constructed based on known stock prices in previous days. The found subgraphs correspond to groups of stocks with similar behavior (in the sense of proximity to the unit of correlation coefficients between sequences of stock prices).

B. A new algorithm of real trading on the stock market, which involves the use of two different schemes depending on earnings in previous days. The proposed algorithm can be called flexible (a well-known term in numerous situations not related to the stock market).

The both algorithms are described in detail in section 3.

Section 4 describes the general trading scheme for 20 years (1995-2014, 80 quarters) with simultaneous parallel use of several identical QTA, whose results of over 20 years are called **variants.**

The experimental results of the proposed general trading scheme are presented in Section 5.

In conclusion possible modifications of the proposed approach and its basic algorithms are specified, as well as some areas of further research and the key findings are given.

2. The S&P 500 stock market and the structure of the quarterly trading algorithm

Let's start with the concepts and definitions necessary for a formal description of the stock market. The paper considers a fragment of the US stock market – S&P-500 (500 largest companies in the USA). Daily data on stock prices at the close of trading was used for 20 years and one month – from 12/01/1994 to 12/31/2014. Individual quarters were considered as the main periods. The number of working days per quarter is in the range of 58-64.

The composition of the shares has changed over the years, but these changes were quite rare (no more than 10 shares left the market per quarter and the same number of new ones appeared, but in most cases, it was 1-4 companies). In addition, all constructions in this section belong to one quarter (there are 80 quarters all in all).

Let t be the fixed working day of a given quarter ($t = 0, 1, 2, ..., p-1$). It seems natural to assume that all stocks traded on the S&P-500 market before day t, as well as their closing prices, are known starting from day x, on l business days preceding day t (including the first business day 0). Therefore, at the beginning of the quarter, the days of the previous quarter are included in the number l of the preceding days. The number l is 15. The choice of the number l, as well as all other parameters used in the proposed general algorithm, will be discussed below, in section 5.1. It is equally natural to assume that stock prices on day t and all subsequent days are unknown.

As stated in the introduction, no assumptions are made about the probabilistic characteristics of stock prices known before t-day. They are considered only as set constants.

The enlarged flowchart in Fig. 1 shows the QTA. The output of the QTA is the income S received in this quarter (it can be negative).

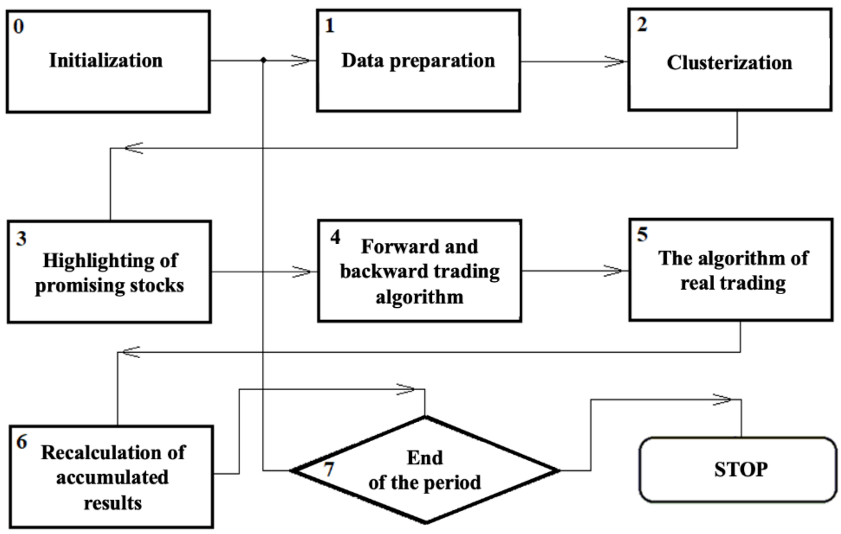


Fig. 1. The enlarged flowchart of the QTA

Before proceeding to the description of the blocks of the presented flowchart, let's take a closer look at the time sequence of operations performed over two business days. This sequence is shown in the Table 1.

Table 1. Time flowchart of the trading algorithm

Trading ends on day t-1	Initialization
	Data preparation
	Clusterization
	Highlighting promising stocks
Trading starts on day t	Forward and backward trading algorithm
	The algorithm of real trading
	Recalculation of accumulated results
Trading ends on day t	Moving on to the next day

The importance of this table lies in the fact that it is the actual sequence of operations performed that is associated with the uncertainty of the stock market. It is not possible to find out the income from operations on day t until all previous operations are completed. And when performing all these operations, the value of the shares at the end of trading is unknown and, therefore, the future gain or loss on that day is unknown. This is discussed in more detail in subsection 3.4. Initialization is performed only on the first business day of the quarter (it is convenient to take its number as 0).

3. Description of the algorithms of the QTA flowchart

Further, the basic algorithms of the QTA flowchart in Fig. 1 are described in subsections 3.0 – 3.7. We will describe them in the same order in which they are executed in the specified flowchart. Let's pay attention to the fact that all the current results of all blocks of the algorithm are saved and can be used in further operations.

3.0. Initialization. The initial day number $t = 0$ and the integer parameter $sg = 1$ are set. Two accumulative scalar quantities $V(t)$ and $S(t)$ are determined. Both at $t = 0$ are initialized with zeros. The number of parallel runs that are performed with the same initial data using the same algorithm is indicated by K. Since the algorithm uses a standard random seed, the results are different. The next three blocks 1 – 3 are executed in parallel for each index i = 1, ..., K. In this paper, K = 5.

3.1. Data preparation. The input for block 1 is the prices of all stocks available on the market (at the time of the termination of trading day before the business day with the number t) for the preceding l business days (see Fig. 1 and Table 1). The output of this block is:

a) The matrix D of the distances between stock prices for l business days (the number $1 - r_{ij}$ is taken as the distance, where r_{ij} is the correlation coefficient between the i-th and j-th stock prices for the last l days). Thus, for correlated stocks, the distance is close to 0, and for strongly dissimilar stocks, it is close to 2.

b) A graph G constructed using this matrix, the vertices of which correspond to the considered set of stocks, and each vertex is connected to the 4 nearest ones by a specified distance. Note that this definition does not imply that the degree of each vertex is 4, but only that it is at least 4.

Here, just for clarity, is a well–known algorithm for constructing this graph, usually called the **neighborhood graph** (in relation to the stock market, this graph is called the **market graph**).

Algorithm for constructing a neighborhood graph

The input of this algorithm is a distance matrix D of size $n \times n$ (n denotes the number of objects in the system under consideration – in this case, the number of shares traded on the day $t-1$ and on the previous 14 days).

Step 1. Let's define an integer matrix A of size $n \times n$ and let all the elements a_{ij} be equal to 0.

Step 2. For each i = 1, ..., n, we perform the following operations.

2.1. Let us determine the indices i_1, i_2, i_3, i_4, k_1..., k_s from the distance matrix D, such that

$$d_{ii_1} \le d_{ii_2} \le d_{ii_3} \le d_{ii_4} = d_{ik_1} = \ldots = d_{ik_s}, \quad (1)$$

and all other elements of the i-th row of the matrix D are strictly larger than those written out in column (1). Thus, objects i_1, i_2, i_3, i_4, k_1..., k_s are the closest to object i. In this case, objects with numbers k_1, ..., k_s are located at the same distance from object i. Note that the very presence of objects k_1, ..., k_s occupying places after the fourth in the chain (1) is not guaranteed:

objects satisfying the required inequalities may simply not exist. At the same time, the existence of objects i_1, i_2, i_3, i_4 occupying the first four places in the chain (1) is guaranteed (in case $n > 4$).

2.2. For all $j = i_1, i_2, i_3, i_4, k_1\ldots, k_s$ let $a_{ij} = 1$ and $a_{ji} = 1$.

The output of the algorithm is the adjacency matrix of the neighborhood graph. It can be clearly seen that the constructed graph does not depend on specific numbering satisfying the above conditions.

3.2. Clusterization. The described algorithm is related to a group of clustering algorithms defined by a certain index of centrality or betweenness of a given undirected graph. In this case, we are talking about the index of the betweenness of edges of the graph. Although the graph decomposition method proposed in [7] deals with a certain index that can be attributed to betweenness indexes, still differs significantly from the algorithms of the mentioned group. Let's look at this in more detail. First, let's briefly outline the Newman-Girvan algorithm [9], one of the first algorithms from the mentioned group.

Each edge of the original graph is assigned an integer value that changes during the algorithm. It can be called an edge load. The operations of the Newman-Girvan algorithm are as follows:

1. Set the current load in each edge to zero.
2. Randomly select two network vertices.
3. Find the path between the vertices found in the previous step. If no such path exists, go to 7; otherwise, go to 4.
4. Add to the loads of all edges included in the path found in step 3, one at a time.
5. If the number of passes of steps 2-4 is large enough (for example, the number of edges with maximum load did not change during the second half of the passes or the number of passes was equal to the preset number), then proceed to the next step 6; otherwise, return to step 2. (The possibility of different implementations of this step is obvious.)
6. Remove an arbitrary edge with maximum load and proceed to step 1.
7. Stop. The network is divided into two connectivity components, which correspond to the desired groups.

The essence of the matter is this. If the graph has two subsets of vertices (groups or communities) connected by a small number of edges, then all shortest paths between vertices from these groups must pass along one of these few edges. Thus, the edges connecting the different groups have a high degree of betweenness. The removal of these edges leads to the separation of groups from each other, and, consequently, to the identification of the desired community structure. The obvious disadvantage of this algorithm is the need to repeatedly return to step 1 when finding a single dichotomy. In this case, all the information found in the previous steps about the accumulated load values in the ribs disappears. This disadvantage is indicated by the authors themselves, along with the remark that removing not only the edge with the maximum load, but also the edges with the next largest loads, can lead to erroneous results.

An alternative approach was proposed in [9], addressing the same indicator in the edge load (referred to therein as frequency) while eliminating the aforementioned limitation. The key idea is that, within the same iteration, the algorithm does not identify a single edge with the maximum load but rather determines a graph cut, in which all edges exhibit the maximum load. Furthermore, after identifying the first dichotomy, it is not necessary to reset all load values – the process continues with the already accumulated loads until the next dichotomy is found, and so forth. As a result, rather than constructing a single dichotomy per iteration, the algorithm generates a family of dichotomies. In simple cases, all these dichotomies coincide, whereas in more complex scenarios (such as those observed in financial markets), that's not so.

The proposed clustering algorithm is described below. Its inputs are a graph *G* and a matrix *D* (see Subsection 3.1). The set of stocks is divided into *r* clusters ($r = 12$). The following provides a detailed description of the clustering algorithm.

1. Initialization. Definition of the required parameters. As stated above, the number of clusters is set to 12. The initial clustering consists of a single element – the original graph *G*. The number of subgraphs is assumed to be 1, and the number of the last subgraph in the clustering under construction is also 1.

2. Selection of the next subgraph from the already constructed family. The subgraph with the maximum number of vertices is selected (arbitrary, if there are several of them).

3. Construction of a family of dichotomies for the subgraph chosen in Step 2. This procedure is described in detail in Subsection 3.2.1.

4. Selection of a dichotomy from the family obtained in Step 3. The selection procedure is discussed in Subsection 3.2.2.

5. Construction of a new clustering. The subgraph selected in Step 2 is replaced by the subgraph from the dichotomy chosen in Step 4 that contains a greater number of vertices. It retains the same index as the replaced subgraph. Additionally, the other subgraph from the dichotomy (identified in Step 4) is incorporated into the clustering and assigned the last index n+1, where n is the number of subgraphs in the previous clustering.

6. Updating the number of subgraphs. The index is updated as n = n+1, ensuring that n now is equal to the number of subgraphs in the newly formed clustering.

7. Verification of the inequality $n < r$. If the condition holds, the algorithm returns to Step 2. Otherwise, the desired clustering is completed. The initial graph G is now partitioned into r disjoint subgraphs, and the algorithm stops.

The output of the presented algorithm consists of three sets:

a) A set of clusters (subsets of the vertex set of the original graph);

b) A set of numerical values representing the density of each constructed cluster;

c) A set of numerical values representing the cardinality of each constructed cluster.

3.2.1. Algorithm for constructing a family of dichotomies (ACFD)

This subsection presents one of the fundamental algorithms used in the proposed general trading algorithm. The ACFD constructs a family of dichotomies F. It consists of an initialization stage and a main stage, which involves T-fold sequential repetition of a general step. The parameter T is one of the predefined parameters of the ACFD, and in this case, T=500. Below, we proceed with its formal description.

A. Initialization. Each edge e_j of a given graph P is assigned a randomly generated integer f_j in the range from 1 to g (inclusive), sampled using a standard generator of uniformly distributed random variables (which is used here for the first time). The parameter g is also predefined within the ACFD, and in this study, it is set to $g = 5$. The value f_j is referred to as the **load** on the corresponding edge (or edge **load**). The maximum of the randomly assigned values f_j is denoted as $F_{\max}$. The output of the initialization stage is a set of load values assigned to all edges of the given graph P. It is important to note that P is one of the subgraphs of the original graph G. During the first execution of the ACFD, the graph P coincides with the graph G.

B. Main Stage. The inputs to the main stage include:

a) The given graph *P*;

b) The current set of load values assigned to all edges of graph *P*;

c) The current maximum value $F_{\max}$.

The output of the main stage is described below, immediately following the description of its steps. The flowchart of the main stage is presented in Fig. 2. The steps of the algorithm at this stage are detailed below.

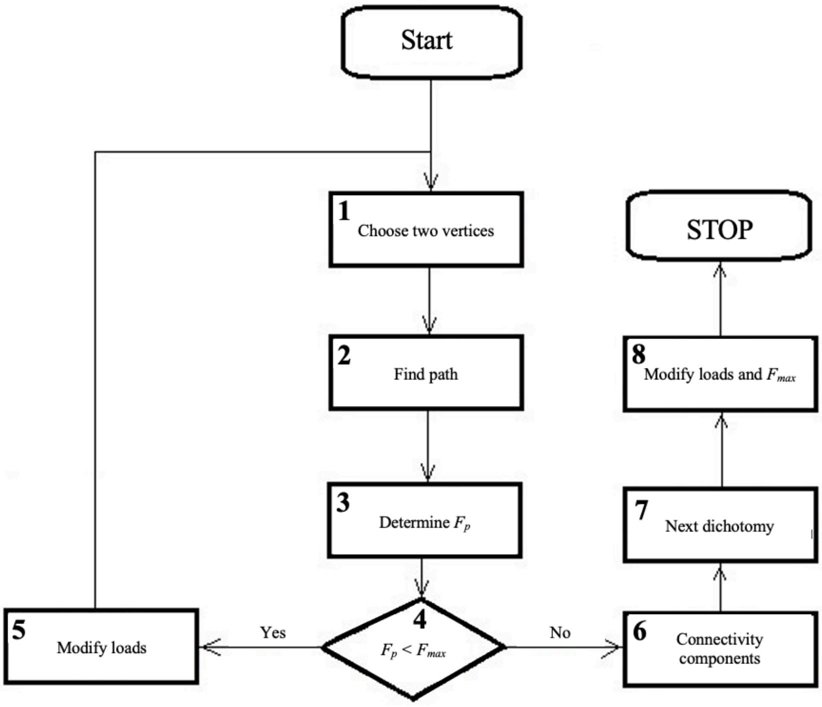


Fig. 2. ACFD flowchart

Steps of the ACFD

1. Using a standard generator of uniformly distributed random variables, two distinct vertices of the graph are selected (this is the second and the last instance of random numbers utilization in the general trading algorithm).

2. Dijkstra algorithm is applied to determine the shortest path connecting the two selected vertices. The length of an edge is defined as its current load. The path length is determined not by summing the edge lengths but by taking the maximum edge length along the path. It is well known that Dijkstra algorithm can be applied in such cases with a single modification: when determining the extended path, instead of summing the length of the initial segment and the newly added edge, the maximum of these two values is recorded. It is important to emphasize that the length of the initial segment is not the sum of its edge loads but the maximum load among those edges.

3. The maximum edge load F_p along the shortest path p found in Step 2 is determined.

4. If $F_p < F_{max}$, proceed to Step 5. Otherwise (i.e., if $F_p \geq F_{\max}$), proceed to Step 6.

5. The edge loads along the path identified in Step 2 are incremented by 1. Then, return to Step 1.

6. The connected components of the given graph P are determined as if all edges with the maximum load were removed. Note that, in reality, no edges are actually removed from the graph, and P remains unchanged.

7.1. The connected component with the maximum number of vertices is designated as the first subgraph of the constructed dichotomy. The union of all remaining components (or the single remaining component, if only one exists) is designated as the second subgraph of the constructed dichotomy.

7.2. When constructing both components, all edges of the original graph are preserved except those connecting vertices from the first component to vertices of the second component. It is evident that this procedure effectively removes a certain minimal cut of the given graph. Consequently, the next dichotomy of graph P is formed. It is also clear that if the original graph was connected, both constructed subgraphs remain connected, too.

8. All edges of the most recently identified path in the original graph are incremented by 1. As a result, at least one of these edges will have a load equal to $F_{\max}+1$. Assign $F_{\max} = F_{\max}+1$.

The current execution of the main stage is now complete. Its outputs are:

a) The modified set of current edge loads in graph P;

b) The updated maximum edge load $F_{\max}$ in graph P;

c) An additional dichotomy of the original graph.

Illustration of Steps 7.1, 7.2, and 8 of the main stage is presented in Figure 3. Below, we provide some explanations and comments on these steps. Prior to executing the comparison of loads in Step 4, three distinct cases − denoted as Cases A, B, and C − are considered. In Fig. 3, bold segments represent edges with the maximum load, while thin lines indicate paths connecting the vertex pair *a* and *b*.

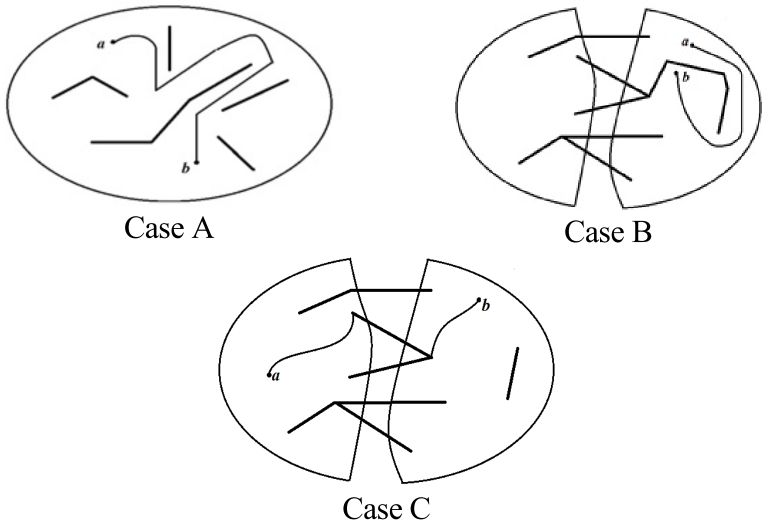


Fig 3. Cuts and paths on the graph

In Case A, the set of all edges with the maximum load does not constitute a cut of the given graph. Consequently, the shortest path found in Step 2 does not contain edges with the maximum load due to the minimax definition of path length. Therefore, the maximum load F_p determined in Step 3 is less than $F_{\max}$, and the algorithm proceeds to Step 5, where the loads of all edges in the identified path are increased by 1. Subsequently, execution returns to Step 1 of the general stage. This consideration is central to the proposed algorithm. Indeed, if the maximum load F_p in the edges of the identified path were equal to $F_{\max}$, this would imply that the set of edges with load $F_{\max}$ forms a cut of the graph, such that the constructed path traverses an edge of this cut. If these edges did not form a cut, the minimax

Dijkstra algorithm would have found a path in which all edges have a load strictly less than $F_{\max}$.

In Case B, the set of all edges with the maximum load does contain a cut of the original graph. However, the identified path does not include edges with the maximum load because its endpoints are located on the same side of the cut. The process thus continues analogously to Case A.

In Case C, the set of all edges with the maximum load constitutes a cut of the original graph, and the endpoints of the identified path are positioned on opposite sides of this cut. Consequently, at least one edge along this path belongs to the specified cut and has a load equal to $F_{\max}$. Therefore, after the comparison in Step 4, the process follows a different trajectory (through Steps 6–8), resulting in the construction of the next dichotomy of the original graph at Step 7.

Note that all the considerations in this subsection can be applied to arbitrary graphs, not just to graphs modeling systems of a specific type. Once again, we emphasize that all edge removals are virtual, and the original graph P remains unchanged throughout all T executions of the main stage.

3.2.2. Selection of a dichotomy. Let us introduce the necessary formal concepts. Let X be a certain set of vertices of the considered graph P, and let n denote the number of vertices in P. Define $D = (d_{ij})$ as the distance matrix between the vertices of P ($i, j = 1, \ldots, n$). We assume that $d_{ij} \geq 0$, $d_{ij} = d_{ji}$ и $d_{ii} = 0$, however, the triangle inequality is not required. The elements of matrix D are interpreted as the degree of dissimilarity between the objects corresponding to vertices i and j of graph P. Let $S(X)$ denote the average distance between the vertices within the set X. Formally,

$$S(X) = \frac{1}{0.5 \times p \times (p+1)} \Sigma d_{ij}, \tag{2}$$

where summation is taken over all unordered pairs of distinct vertices from the set X, and $p=|X|$. Additionally, we define $S(X)=0$ for single-element sets X.

It is conceptually evident that the function $S(X)$ characterizes the density of objects within the set X and their mutual proximity. A lower value of $S(X)$ indicates a higher degree of similarity among the vertices in X, and, consequently, among the objects they represent within the system modeled by graph P.

Now, we can formulate the criterion for selecting a single dichotomy from the constructed set of dichotomies. Each dichotomy consists of two subgraphs with vertex sets X_1 and X_2. Let us consider the densities $S(X_1)$ and $S(X_2)$. As the quality criterion of a dichotomy (X_1, X_2), we take the larger (i.e., the worst-case) value among the two numbers − $S(X_1)$ and $S(X_2)$. Accordingly, the optimal dichotomy among all available dichotomies is the one that minimizes this criterion.

Let us draw attention to the fact that the assumption of setting distance matrix is used only at a single stage of the ACFD process − the stage of selecting one dichotomy from the constructed set. This naturally raises the question of what to do if the distance matrix is not provided. Essentially, some reasonable characteristic of graph density is required. Such a characterristic can be taken as the ratio of the number of edges in the graph to their maximum possible number, $0.5\times(n-1)\times n$, and then applying a minimax approach − i.e., selecting the worst (minimum) density value from the two subgraphs forming the dichotomy and then choosing the dichotomy for which this worst value is maximized. Finally, one can select the minimum number of vertices across the two subgraphs and then choose the dichotomy for which this value is maximized. This method expresses the desire to divide the graph into two parts as "equally" as possible.

In relation to the financial market considered in this paper, a well-known approach is naturally applicable: calculating the correlation coefficients r_{ij} between the price time series of two stocks i and j and determining the distance between them using formula (1).

Thus, the output of Block 2 (clustering) in the flowchart of the QTA (Fig. 2) consists of:

a) a set of clusters;

b) the mean pairwise distances between elements within each cluster;

c) the number of elements in each cluster.

It should be emphasized that we are dealing with K realizations of these sets. All of them collectively serve as input to Block 3, where the algorithm determines the best cluster within each realization.

3.3. Highlighting promising sets of shares. The following is done in each of the K cluster groups.

1. All clusters with fewer than 20 or more than 200 elements (shares) or an average distance greater than 0.6 (with a possible maximum of 2) are deleted.

2. The remaining clusters in the group are compared according to two criteria: the number of elements (maximization) and the average distance

(minimization). The optimal cluster is selected using the ideal point method.

Therefore, there are K sets of shares that are considered promising (see subsection 3.4 below). If there are simply no clusters satisfying the conditions of item 1 in the i-th group, then the i-th set of promising stocks is declared empty. This fact is taken into account in blocks 5 and 6 when calculating the one-day gain or loss. Such situations actually occur in many neighborhoods. The situation when for some days t all perspective sets turn out to be empty is not excluded.

Note that the same stocks may be included in different prospective sets.

3.4. Choosing a trading algorithm. This section is mandatory before moving on to forward trading (see Table 1). Before discussing the operations of Block 4, which is one of the key QTA blocks, it is necessary to introduce the relevant concepts.

3.4.1. Forward and backward algorithms of daily trading. The main idea in developing various stock market trading algorithms was to assume that the processes were repeatable for at least one day. A certain set of stocks with fairly similar behavior is determined for several days preceding day t (which can be called a set of promising stocks) and it is assumed that the prices of promising stocks will change monotonously. Formally, this means that one of three situations is possible for each promising stock:

$$\text{A: } c(t\text{–}2) > c(t\text{–}1),$$
$$\text{B: } c(t\text{–}2) < c(t\text{–}1),$$
$$\text{C: } c(t\text{–}2) = c(t\text{–}1),$$

and at the same time, the direction of price change in cases A and B will remain the same, i.e. in case A it turns out that

$$c(t\text{–}2) > c(t\text{–}1) > c(t), \tag{3a}$$

and in case B, it turns out that

$$c(t\text{–}2) < c(t\text{–}1) < c(t), \tag{3b}$$

where $c(t)$ denotes the stock price at market closing on day t.

It is not assumed that the second inequalities in (3a) and (3b) will be fulfilled for all promising stocks, but it is assumed that they will be fulfilled for **most** of them.

Without discussing the various methods of determining promising stocks, let's pay special attention to the fact that the values of $c(t\text{–}2)$ and $c(t\text{–}1)$ at the close of trading on day t–1 are known, but the price of $c(t)$ is naturally not. This is **the uncertainty of the stock market**.

The trading algorithm, which can be called **forward**, is connected with the simple reasoning carried out. Each stock from group A is sold at the opening of the market on day t and is bought at the closing of the market on day t at the price $c(t)$, which will be known on day t. Each stock from the group B is bought at the market opening on day t and sold at the market closing on day t at the price $c(t)$, which will be known on day t. For stocks from group C (which are very few or none at all) nothing is being done.

Indeed, in most cases, a forward algorithm generates revenue. However, not always. Let's look at these situations in more detail. Let inequalities (3a) and (3b) be replaced by inequalities (4a) and (4b):

$$c(t-2) > c(t-1) < c(t), \tag{4a}$$

$$c(t-2) < c(t-1) > c(t). \tag{4b}$$

A trading algorithm can be associated with this situation, which is naturally called an **backward** one. Each stock from group A is bought at the opening of the market on day t and sold at the closing of the market on day t at the price $c(t)$, which will be known on day t. Each stock from group B is sold at the market opening on day t and bought at the market closing on day t at the price $c(t)$, which will be known on day t.

A simple fact immediately follows from formulas (3) and (4).

Statement 1. Under any circumstances, the total gain from the forward and backward algorithms is always zero.

At first glance, inequalities (4) might seem relatively rare. However, this is not the case. In complex stock market conditions, the price evolution process becomes chaotic. Drawing on a hydraulic analogy, one could say that inequalities (3) correspond to waves, while inequalities (4) give rise to so-called "dead swell," which is more dangerous for ships than even large waves. Negative outcomes when applying the forward algorithm occur quite frequently; moreover, the absolute magnitude of losses in these instances far exceeds the gains observed during more regular intervals. All these ef-fects will be demonstrated using real data from the S&P-500 stock market in subsection 5.2.

Of course, we do not know in advance which scenario will unfold on any given day. Nevertheless, it is possible to propose a certain heuristic, whose sole justification is the consistently positive results that it has produ-ced over the 20-year period described in section 2. Experimental findings are provided in section 5.

Before proceeding further, we will illustrate the application of both al-gorithms in simple model scenarios.

Example 1. Demonstration of gains and losses under the forward and in-verse algorithms.

Case 1. Let the price of a certain stock be $c(t-2)=5$, $c(t-1)=3$, $c(t)=2$.

Forward algorithm. The stock is sold on the morning of day t at the previous evening price of 3 and bought at the current evenings price of 2. The stock remains in the same hands, yielding a profit of +1.

Backward algorithm. The stock is bought on the morning of day t at the previous evening price of 3 and sold at the current evening price of 2. The profit in this case is −1.

Case 2. Let the price of a certain stock be $c(t-2)=3$, $c(t-1)=4$, $c(t)=6$.

Forward algorithm. The stock is bought on the morning of day t at the pre-vious evening price of 4 and sold at the current evening price of 6. The profit is +2.

Backward algorithm. The stock is sold on the morning of day t at the pre-vious evening price of 4 and then bought at the current evening price of 6. The stock remains in the same hands, resulting in a profit of −2.

Case 3. Let the price of a certain stock be $c(t-2)=5$, $c(t-1)=2$, $c(t)=3$.

Forward algorithm. The stock is sold on the morning of day t at the previous evening price of 2 and bought at the current evening price of 3. The stock remains in the same hands, and the profit is −1.

Backward algorithm. The stock is bought on the morning of day t at the previous evening price of 2 and sold at the current evening price of 3. The profit in this scenario is +1.

Case 4. Let the price of a certain stock be $c(t-2)=2$, $c(t-1)=5$, $c(t)=3$.

Forward algorithm. The stock is bought on the morning of day t at the pre-vious evening price of 5 and sold at the current evening's price of 3. The profit is −2.

Backward algorithm. The stock is sold on the morning of day t at the pre-vious evening price of 5 and then bought at the current evenings price of 3. The stock remains in the same hands, resulting in a profit of +2.

3.4.2. Description of the proposed heuristic (Block 4). Recall that by the start of day t, the accumulated quantities $V(t)$ and $S(t)$ are known for $t-1, t-2, \ldots, 0$ (these are calculated in Block 6 immediately after the completion of each day operations). Consider the four most recent known values $V(t-1)$, $V(t-2)$, $V(t-3)$, $V(t-4)$ and check the following four inequalities:

$$V(t-1) < -100,\ V(t-2) < -100,\ V(t-3) < -100,\ V(t-4) < -100. \quad (5)$$

If all these inequalities are satisfied, then **from day *t* until the end of the quarter**, the backward algorithm will be used to determine the trading algo-rithm, and condition (5) will no longer be checked. Otherwise, on day t, the forward algorithm will be executed. Subsequently, at each step until the end of the quarter, a similar check will be performed under the same out-puts. Note that the same value sg is used for all promising sets available on a given day.

Formally, this procedure is expressed by modifying the quantity sg defined in Block 0. If conditions (5) are satisfied, sg is set to -1 for the remainder of the quarter; otherwise, sg=1. The value of sg is the output of Block 4 of the algorithm under consideration. In determining sg, the values of $V(t)$ previously obtained in Block 6 are used.

The rationale behind the proposed heuristic is that if four consecutive negative values arise under the forward algorithm, it may be assumed that subsequent negative returns will predominantly decrease. By virtue of Statement 1 in subsection 3.4.1, this implies that returns from the backward algorithm will increase accordingly. Examples using real data will be presented in Section 5.

3.5. The real trading algorithm. The inputs to this algorithm, whose flowchart is shown in Fig. 1, are K sets of promising stocks identified in Block 3, along with the value of sg obtained in Block 4. For each non-empty set of promising stocks, the operations described at the beginning of subsection 3.4 are performed:

When *sg*=1, each stock from group A is sold at the market opening on day t and repurchased at the market close on day t at the price $c(t)$, which is determined on day t. Each stock from group B is purchased at the market opening on day t and sold at the market closing on day t at the price $c(t)$, which is determined on day t.

When *sg*=−1, each stock from group A is purchased at the market opening on day t and sold at the market close on day t at the price $c(t)$. Each stock from group B is sold at the market opening on day t and repurchased at the market close on day t at the price $c(t)$.

The output of Block 5 is the collection of K stock sets together with the respective purchase and sale prices for each stock at the opening and closing of trading on day t.

3.6. Recalculation of accumulated gains/losses. In Block 6, the first step is a straightforward operation that sums the gains/losses realized from actual trading on day t (see Block 5). For empty sets of prospective stocks, no calculation is performed. The gains/losses from each stock are counted as many times as that stock appeared in the identified non-empty promising sets. Denote the resulting sum by $U(t)$. Let us return to the quantities $V(\tau)$ and $S(\tau)$ introduced in Block 1. These have already been determined for all $\tau<t$ (recall that for $\tau=0$, they are initialized to zero in Block 0).

We set:

$$V(t) = V(t-1) + U(t),\ S(t) = S(t-1) + U(t),\ \text{if } sg = 1, \qquad (6a)$$

$$V(t) = V(t-1) - U(t),\ S(t) = S(t-1) + U(t),\ \text{if } sg = -1. \qquad (6b)$$

An important clarification is in order. The values $S(t)$ represent the actual profits/losses obtained using the proposed QTA. In contrast, the values $V(t)$ are purely virtual. They correspond to the profits/losses that would have been obtained if only the forward trading algorithm been used. Note that the signs for the new values of $S(t)$ coincide. This arises from the fact that the gains $U(t)$ themselves are calculated using different formulas.

3.7. End of the Period. In Block 7, which differs from the other blocks in that it is a logic block, the system either proceeds to the next day t+1 or terminates the QTA. In the latter case, the real profit $S(t)$ accumulated over the quarter constitutes the output of the QTA.

4. General trading scheme

The QTA is applied sequentially to each of the 80 quarters. As noted earlier, for each quarter and each trading day, only the stock price data for preceding days and certain already computed quantities (see formulas (5) and (6)) are used. The trading results for each quarter, for each year, and the principal outcome − the sequence of accumulated annual returns − are collectively referred to as a variant. These variants are denoted by letters a, b, c, d, …. Recall that all variants were computed using the same initial data (i.e., daily closing stock prices) and the same software. The difference in the results is determined solely by multiple use of a standard random number generator when selecting vertex pairs as input to Dijkstra algorithm (see subsection 3.2.1).

5. Experimental results

5.1. Parameter selection. It should be noted that the developed algorithms make use of numerous parameters, as mentioned in the description of the QTA. Their selection is justified by little more than common sense. This is because the present work makes no attempt to simulate the processses of price formation or trading on the stock market; Instead, all the algorithms analyze already known, fixed real-world data from previous days. The only justification for choosing one set of parameters over another is the potential for obtaining positive outcomes. Some experimental results − subject to natural qualifications − can be regarded as favorable. These are presented in Section 5.

5.2. Examples of the flexible algorithm in operations. Below we present detailed experimental data for variant **a**. First, let us consider in which quarters the flexible algorithm was actually applied, i.e., in which quarters inequalities (5) were satisfied. The use of the backward algorithm can either increase or decrease the profit for a given quarter compared to the forward algorithm.

The quarters that produced negative or positive outcomes (referring solely to the sign of the difference between the results of the backward and forward algorithms) are listed in Table 2.

Table 2. Use of the flexible algorithm

Sign	Number of quarters	List
−	15	1995_2, 1997_1, 1998_1, 1998_3, 1998_4, 2001_2, 2002_2, 2003_1, 2003_4, 2004_1, 2006_4, 2010_1, 2011_2, 2011_3, 2012_2
+	33	1998_1, 1999_4, 2000_2, 2000_4, 2002_4, 2003_2, 2003_3, 2004_3, 2004_4, 2005_2, 2005_3, 2005_4, 2013_1, 2007_1, 2007_2, 2007_3, 2007_4, 2008_1, 2008_2, 2008_3, 2009_1, 2009_2, 2009_4, 2010_2, 2010_3, 2010_4, 2011_1, 2011_4, 2013_1, 2013_2, 2013_4, 2014_1, 2014_3

Note that in the remaining 32 quarters (out of 80), the backward algorithm was not used.

We now provide examples of value accumulation in selected quarters from the positive group. It is worth noting the various sign combinations of the accumulated sum (apart from the case of positive virtual accumulation and negative actual accumulation). By construction, such a scenario can only occur in quarters belonging to the negative group. The days on which a transition from the forward algorithm to the backward algorithm took place are underlined in Table 3.

Table 3. Examples of switching between algorithms

2003_3				2007_1			
0	0	0	0	0	0	0	0
1	55.39	55.40	55.40	1	19.84	19.84	19.84
2	-72.77	-17.38	-17.38	2	3.30	23.14	23.14
3	-200.98	-218.36	-218.36	3	-3.34	19.80	19.80
4	69.23	-149.13	-149.13	4	22.72	42.52	42.52
5	-14.06	-163.19	-163.19	5	93.32	135.84	135.84
6	81.46	-81.72	-81.72	6	40.27	176.11	176.11
7	-47.68	-129.41	-129.41	7	1.72	177.83	177.83
8	57.02	-72.39	-72.39	8	-24.80	153.03	153.03
9	-37.41	-109.80	-109.80	9	4.60	157.63	157.63
10	40.58	-69.22	-69.22	10	35.10	192.73	192.73
11	55.60	-13.61	-13.61	11	-28.44	164.29	164.29
12	-103.93	-117.54	-117.54	12	-48.89	115.40	115.40
13	-81.01	-198.55	-198.55	13	-66.46	48.94	48.94
14	-55.10	-253.66	-253.66	14	-45.60	3.34	3.34
15	**-51.38**	**-305.05**	**-305.05**	15	-178.24	-174.90	-174.90
16	-29.80	-334.85	-275.24	16	5.63	-169.28	-169.28
17	-49.23	-384.08	-226.01	17	-0.62	-169.89	-169.89
18	3.79	-380.30	-229.80	**18**	**1.14**	**-168.75**	**-168.75**
19	4.14	-376.15	-233.94	19	46.11	-122.64	-214.86
20	1.57	-374.59	-235.51	20	47.90	-74.73	-262.76
21	0.28	-374.31	-235.79	21	30.53	-44.21	-293.29
22	-22.38	-396.68	-213.41	22	-15.34	-59.55	-277.95
23	-2.04	-398.72	-211.38	23	0.27	-59.28	-278.22
24	3.00	-395.72	-214.38	24	60.50	1.22	-338.71
25	-0.39	-396.11	-213.99	25	-29.20	-27.98	-309.51
26	-8.55	-404.65	-205.44	26	12.21	-15.77	-321.73
27	18.51	-386.15	-223.95	27	24.06	8.29	-345.78
28	-1.76	-387.91	-222.19	28	-29.72	-21.44	-316.06
29	19.66	-368.25	-241.84	29	110.33	88.89	-426.39
30	-24.27	-392.52	-217.58	30	31.46	120.36	-457.85

31	-4.28	-396.80	-213.29	31	-20.32	100.04	-437.54
32	-18.84	-415.64	-194.45	32	25.94	125.98	-463.48
33	-13.48	-429.12	-180.97	33	-112.26	13.72	-351.21
34	44.68	-384.44	-225.65	34	3.19	16.90	-354.40
35	4.38	-380.06	-230.03	35	25.24	42.15	-379.64
36	12.14	-367.93	-242.17	36	31.44	73.58	-411.08
37	-76.43	-444.36	-165.74	37	106.36	179.94	-517.44
38	34.29	-410.07	-200.03	38	-80.72	99.22	-436.72
39	0.06	-410.01	-200.09	39	-44.77	54.45	-391.95
40	-1.64	-411.65	-198.44	40	115.45	169.90	-507.40
41	39.54	-372.11	-237.98	41	305.51	475.41	-812.91
42	57.71	-314.41	-295.69	42	-419.52	55.89	-393.39
43	90.75	-223.66	-386.44	43	-111.83	-55.93	-281.56
44	24.23	-199.43	-410.67	45	25.45	-106.74	-230.75
45	-27.08	-226.50	-383.59	46	0.00	-106.74	-230.75
46	39.65	-186.86	-423.24	47	-211.47	-318.22	-19.28
47	-58.98	-245.84	-364.25	48	-166.62	-484.84	147.34
48	-69.64	-315.48	-294.62	49	79.61	-405.23	67.73
49	119.56	-195.91	-414.18	50	-96.14	-501.37	163.87
50	-63.61	-259.52	-350.57	51	-94.26	-595.63	258.13
51	13.02	-246.50	-363.60	52	59.04	-536.59	199.09
52	-0.38	-246.88	-363.21	53	147.68	-388.91	51.41
53	-74.14	-321.03	-289.07	54	-25.28	-414.19	76.69
54	-14.06	-335.09	-275.00	55	0.39	-413.80	76.30
55	-59.55	-394.64	-215.45	56	6.11	-407.69	70.19
56	-1.46	-396.11	-213.99	57	31.74	-375.95	38.45
57	63.70	-332.40	-277.69	58	134.71	-241.24	-96.26
58	-54.26	-386.66	-223.44	59	-61.34	-302.58	-34.92
59	-97.76	-484.41	-125.68	60	-67.15	-369.73	32.23
60	55.80	-428.61	-181.48				
61	82.44	-346.18	-263.92				
62	-67.65	-413.82	-196.27				
63	-70.00	-483.82	-126.27				

2008_3				2011_4			
0	0	0	0	0	0	0	0
1	73.17	73.17	73.17	1	-824.32	-824.32	-824.32
2	-41.09	32.08	32.08	2	247.09	-577.23	-577.23
3	5.86	37.95	37.95	3	361.88	-215.35	-215.35
4	0.00	37.95	37.95	**4**	**-339.87**	**-555.22**	**-555.22**
5	-729.20	-691.25	-691.25	5	-427.53	-982.75	-27.69
6	-132.32	-823.58	-823.58	6	2.53	-980.22	-30.22
7	-24.97	-848.54	-848.54	7	-6.10	-986.32	-24.12
8	**229.02**	**-619.52**	**-619.52**	8	-33.19	-1019.50	-90.93

9	97.31	-522.22	-716.83	9	-212.70	-1232.20	121.77
10	-240.99	-763.21	-475.84	10	-286.47	-1518.67	408.23
11	664.27	-98.94	-1140.11	11	-586.64	-2105.31	994.88
12	-41.81	-140.75	-1098.31	12	-347.13	-2452.44	1342.01
13	-23.41	-164.16	-1074.89	13	-168.37	-2620.82	1510.38
14	-186.12	-350.27	-888.78	14	388.70	-2232.11	1121.68
15	176.02	-174.25	-1064.80	15	0.00	-2232.11	1121.68
16	-563.39	-737.64	-501.41	16	-416.53	-2648.64	1538.21
17	29.37	-708.27	-530.78	17	-286.30	-2934.95	1824.51
18	-10.28	-718.55	-520.50	18	434.81	-2500.14	1389.70
19	-477.57	-1196.12	-42.93	19	-14.77	-2514.91	1404.47
20	96.52	-1099.60	-139.46	20	-183.02	-2697.93	1587.49
21	-125.61	-1225.20	-13.85	21	740.47	-1957.46	847.02
22	52.84	-1172.37	-66.68	22	-453.23	-2410.68	1300.25
23	57.92	-1114.45	-124.60	23	528.46	-1882.23	771.79
24	-4.23	-1118.68	-120.37	24	-89.06	-1971.29	860.85
25	19.06	-1099.62	-139.43	25	-56.72	-2028.00	917.57
26	-38.93	-1138.55	-100.50	26	46.55	-1981.45	871.01
27	-322.29	-1460.83	221.78	27	-722.88	-2704.33	1593.89
28	158.88	-1301.95	62.90	28	-152.51	-2856.84	1746.40
29	-84.93	-1386.88	147.82	29	247.57	-2609.27	1498.83
30	14.18	-1372.70	133.65	30	-168.99	-2778.26	1667.82
31	-160.47	-1533.16	294.11	31	-68.64	-2846.90	1736.46
32	123.65	-1409.51	170.46	32	-119.63	-2966.53	1856.10
33	-232.76	-1642.27	403.22	33	192.70	-2773.83	1663.40
34	264.68	-1377.59	138.54	34	8.27	-2765.56	1655.13
35	74.79	-1302.80	63.75	35	-13.22	-2778.79	1668.35
36	-1.05	-1303.85	64.80	36	59.36	-2719.43	1608.99
37	-19.63	-1323.47	84.42	37	-0.21	-2719.64	1609.20
38	-201.26	-1524.74	285.69	38	2.69	-2716.95	1606.52
39	-29.53	-1554.26	315.21	39	0.00	-2716.95	1606.52
40	19.30	-1534.96	295.91	40	0.00	-2716.95	1606.52
41	155.18	-1379.78	140.73	41	43.91	-2673.04	1562.61
42	-174.29	-1554.07	315.02	42	-58.64	-2731.68	1621.25
43	-38.74	-1592.80	353.75	43	9.32	-2722.36	1611.92
44	54.28	-1538.53	299.48	44	9.54	-2712.82	1602.39
45	-295.99	-1834.51	595.46	45	-41.62	-2754.44	1644.01
46	-219.47	-2053.99	814.94	46	-6.42	-2760.86	1650.43
47	-5.24	-2059.23	820.18	47	-131.31	-2892.17	1781.74
48	-251.12	-2310.36	1071.31	48	-191.43	-3083.60	1973.16
49	-114.83	-2425.19	1186.14	49	0.00	-3083.60	1973.16
50	140.81	-2284.38	1045.33	50	187.49	-2896.10	1785.67
51	32.28	-2252.10	1013.05	51	0.00	-2896.10	1785.67
52	-546.24	-2798.34	1559.29	52	-18.22	-2914.32	1803.88
53	-384.09	-3182.43	1943.38	53	60.84	-2853.48	1743.04
54	-230.89	-3413.32	2174.27	54	-40.26	-2893.74	1783.31
55	-475.68	-3889.00	2649.94	55	-637.00	-3530.75	2420.31

56	1002.92	-2886.08	1647.03	56	-23.09	-3553.84	2443.41
57	-339.10	-3225.18	1986.13	57	-5.72	-3559.56	2449.13
58	89.15	-3136.03	1896.98	58	40.24	-3519.32	2408.89
59	-106.24	-3242.27	2003.22	59	8.80	-3510.53	2400.09
60	27.52	-3214.75	1975.69	60	-150.79	-3661.32	2550.89
61	96.61	-3118.13	1879.08	61	-219.53	-3880.85	2770.42
62	-372.78	-3490.92	2251.87	62	-173.38	-4054.24	2943.80
63	-476.44	-3967.35	2728.30				

5.3. Accumulated sums for the 1995–2014 period. A complete overview of the quarterly, annual, and cumulative results over the 20-year span for variants ***a***, ***b***, ***c***, ***d*** is presented in Tables 4. Despite minor negative outcomes in certain quarters and even some entire years, the final column demonstrates a pattern of rapid growth in all the variants.

Table 4. Trading results of flexible algorithm for four variants
Variant a

	1 quarter	2 quarter	3 quarter	4 quarter	Year income	Accum. income
1995	194	-201	193	25	211	211
1996	417	146	554	385	1502	1713
1997	-1331	-165	266	848	-382	1331
1998	345	1447	-1487	-782	-477	854
1999	94	268	295	250	907	1761
2000	291	727	809	84	1911	3672
2001	-384	-829	1617	1310	1714	5386
2002	608	-975	1206	1429	2268	7654
2003	-430	137	-126	-196	-615	7039
2004	-640	184	73	179	-204	6835
2005	-234	-99	327	254	248	7083
2006	417	541	5	-279	684	7767
2007	32	156	1317	330	1835	9602
2008	2259	1048	2728	911	6946	16548
2009	-52	-261	1128	-34	781	17329
2010	-609	281	439	-227	-116	17213

2011	-60	-821	-1803	2943	259	17472
2012	203	-400	1152	-314	641	18113
2013	32	-193	773	-439	173	18286
2014	-2	157	-17	577	715	19001

Variant b

	1 quarter	2 quarter	3 quarter	4 quarter	Year income	Accum. income
1995	145	-285	128	150	138	138
1996	340	153	630	184	1307	1445
1997	-1467	51	125	876	-415	1030
1998	46	1339	-1234	-951	-800	230
1999	177	-464	423	107	243	473
2000	561	1138	622	-182	2139	2612
2001	-672	-1313	1304	1924	1243	3855
2002	792	-734	1111	1123	2292	6147
2003	763	138	-58	-351	492	6639
2004	-322	271	163	-106	6	6645
2005	-157	-74	199	170	138	6783
2006	641	-97	5	-329	220	7003
2007	-704	-312	1243	446	673	7676
2008	1568	960	4177	43	6748	14424
2009	2760	-234	792	115	3433	17857
2010	-501	335	140	101	75	17932
2011	-363	-1161	-2054	3615	37	17969
2012	-1	-168	936	-306	461	18430
2013	-209	-153	578	438	654	19084
2014	33	-47	-85	965	866	19950

Variant c

	1 quarter	2 quarter	3 quarter	4 quarter	Year income	Accum. income
1995	142	-477	-301	116	-520	-520
1996	193	143	561	190	1087	567
1997	209	-505	65	643	412	979
1998	288	962	973	-160	2063	3042
1999	53	149	-424	414	192	3234
2000	272	1064	879	-1223	992	4226
2001	-11	-1311	1424	1433	1535	5762
2002	745	-750	1395	1600	2990	8752
2003	-403	416	-41	-626	-654	8098
2004	-686	136	45	-193	-698	7400
2005	-171	-162	254	54	-26	7374
2006	298	326	-632	94	86	7460
2007	-314	-59	910	599	1136	8596
2008	1829	1342	2887	686	6744	15340
2009	1960	835	-446	-666	1683	17032
2010	293	714	331	-237	1101	18124
2011	-247	-1477	-93	3058	1241	19365
2012	-150	303	1108	-1066	195	19560
2013	103	-526	-338	-472	-1233	18327
2014	237	432	-264	13	418	18747

Variant d

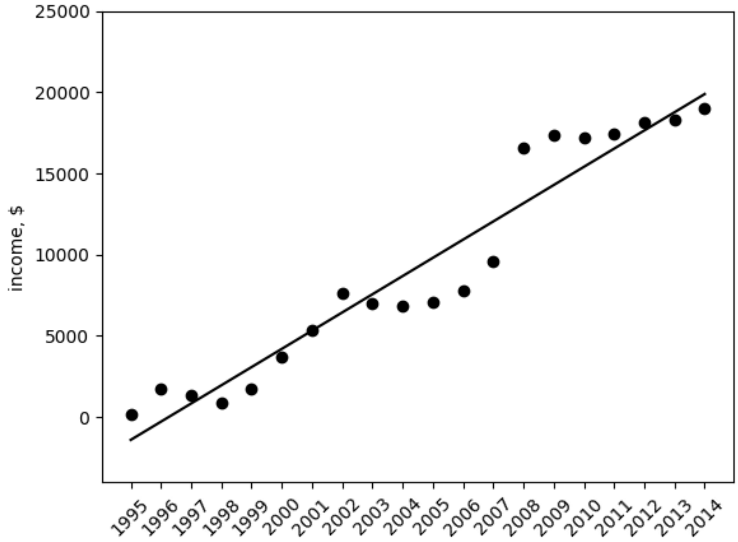
	1 quarter	2 quarter	3 quarter	4 quarter	Year income	Accum. income
1995	184	-293	-232	-19	-360	-360
1996	345	44	795	66	1250	890
1997	902	-343	-6	1093	1646	2546
1998	207	1007	4	-104	1114	3660
1999	-11	355	465	138	947	4607

2000	723	1279	720	-501	2221	6828
2001	-572	-928	1295	1253	1048	7876
2002	971	-601	1172	1207	2749	10625
2003	36	107	331	-603	-129	10496
2004	-988	328	-767	-259	-1686	8810
2005	-205	-382	376	-267	-478	8332
2006	305	398	101	-329	475	8807
2007	-90	-263	1383	308	1338	10145
2008	1737	1294	2671	441	6143	16288
2009	2779	-234	-153	-139	2253	18541
2010	313	646	661	208	1828	20369
2011	-240	-1338	575	3283	2280	22649
2012	-203	455	1183	-256	1179	23828
2013	-193	-48	438	-277	-80	23748
2014	139	-310	597	-391	35	23783

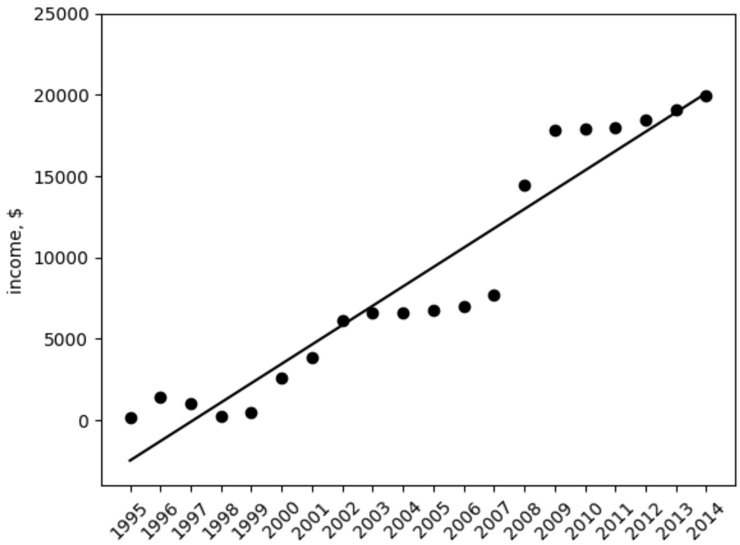
The data in Tables 4a–4d point to a similar pattern in the evolution of accumulated returns across different calculation variants. Once again, it should be emphasized that the quantitative results presented in these tables were obtained using the same software and the same input data. The only differences stem from the use of a standard random number generator in the clustering algorithm.

A visual representation of these results is provided in Fig. 4a – 4d. The summary visual data for all four variants are presented in Fig. 5. Table 5 presents the general quantitative data (in contrast to Tables 4, quarterly data are not included). The variants can be implemented simultaneously, and the overall profit is the sum of the individual profits from these variants.

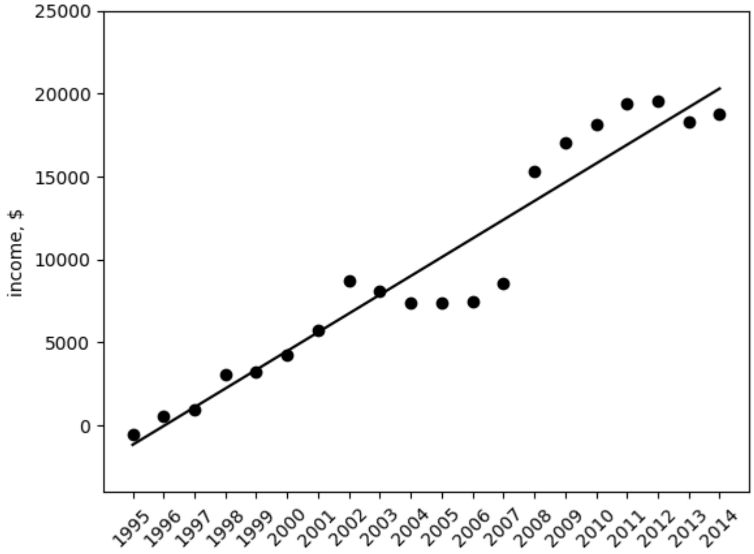
In Table 6, the values of the correlation coefficients are given for the last five columns of Table 5, i.e., for the accumulated incomes of variants **a**, **b**, **c**, **d** and the sum of them.


25000
20000
15000
10000
5000
0
income, $
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014

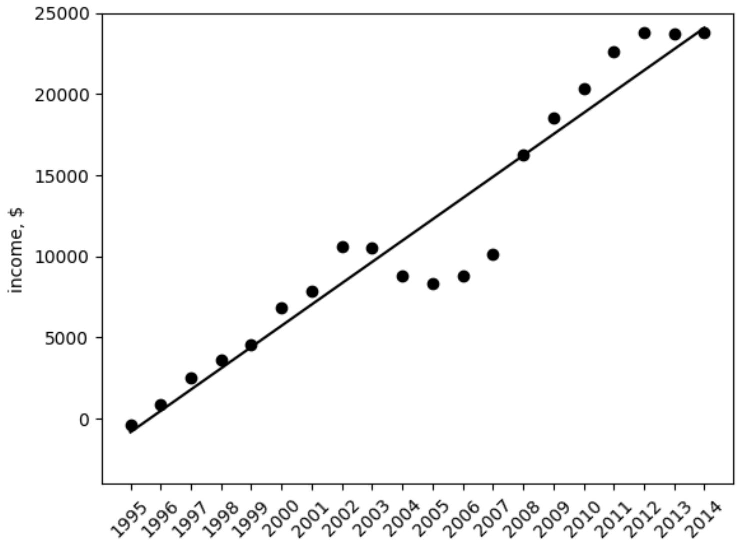
a


25000
20000
15000
10000
5000
0
income, $
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014

b



c



d

Fig. 4. Accumulated totals and regression lines for 4 variants

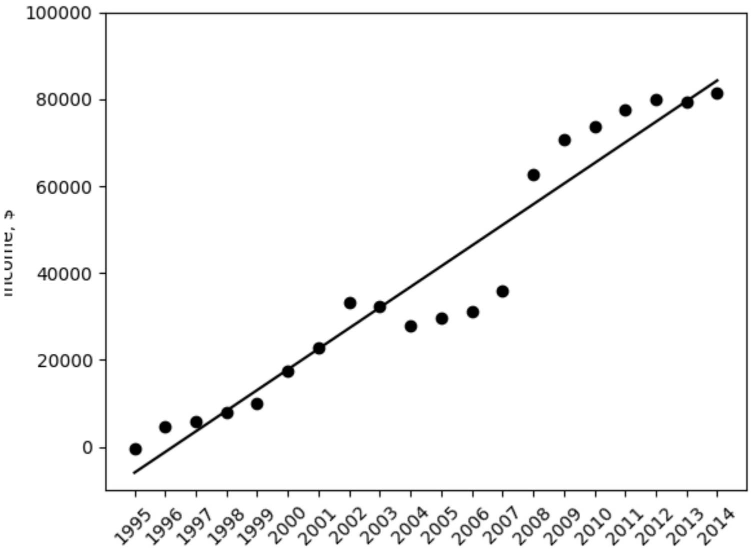


Fig. 5. Visual data for the sum of the variants

Table 5. Comparison of the accumulated returns for variants a, b, c, d, and their sum

Year	Inc. a	Inc. b	Inc. c	Inc. d	Ac. a	Ac. b	Ac. c	Ac. d	Ac. Sum
1995	211	138	-520	-360	211	138	-520	-360	-531
1996	1502	1307	1087	1250	1713	1445	567	890	4615
1997	-382	-415	412	1646	1331	1030	979	2546	5886
1998	-477	-800	2063	1114	854	230	3042	3660	7786
1999	907	243	192	947	1761	473	3234	4607	10075
2000	1911	2139	992	2221	3672	2612	4226	6828	17338
2001	1714	1243	1535	1048	5386	3855	5762	7876	22879
2002	2268	2292	2990	2749	7654	6147	8752	10625	33178
2003	-615	492	-654	-129	7039	6639	8098	10496	32272
2004	-204	6	-698	-1686	6835	6645	7400	8810	27852
2005	248	138	-26	-478	7083	6783	7374	8332	29572

2006	684	220	86	475	7767	7003	7460	8807	31037
2007	1835	673	1136	1338	9602	7676	8596	10145	36019
2008	6946	6748	6744	6143	16548	14424	15340	16288	62600
2009	781	3433	1683	2253	17329	17857	17032	18541	70759
2010	-116	75	1101	1828	17213	17932	18124	20369	73638
2011	259	37	1241	2280	17472	17969	19365	22649	77455
2012	641	461	195	1179	18113	18430	19560	23828	79931
2013	173	654	-1233	-80	18286	19084	18327	23748	79445
2014	715	866	418	35	19001	19950	18745	23783	81479

Table 6. Correlation coefficients among variants a, b, c, d and their sum

	a	b	c	d	sum
a	1	0.994	0,990	0.978	0.995
b		1	0.986	0.979	0.994
c			1	0.992	0.997
d				1	0.993
sum					1

From Tables 4 – 6, it is clear that “similarity” increases with the level of aggregation. A comparison of cumulative results over a prolonged period demonstrates a degree of stability that, at first glance, might be unexpected in an inherently uncertain environment such as the stock market. Nevertheless, the rise in stability with greater levels of aggregation is a well-known systemic effect. As an example, one might point to temperature variations across large regions: on individual days and at specific locations, no discernible patterns are apparent. However, when examining longer time spans and broader geographic areas, stability increases − reaching as much as 12 °C for extended periods on a global average. It is important to note a significant distinction, however. The claim here does not pertain to the stability of the S&P 500 stock market per se; rather, it concerns ***the stability of trading outcomes derived from the proposed algorithm***, applied to actual daily stock prices.

Conclusion

1. Let us begin with the central question − how were the positive results obtained, and what is the proposed explanation? One might say this question reduces to another: why, in the overwhelming majority of cases, do we obtain reasonable clusterings of comparable sizes? For example, with volumes such as:

133 1 1 1 36 33 1 64 58 9 160 1,

whereas sometimes (albeit very rarely) one observes highly uneven clusterings, for example:

477 1 1 2 7 1 1 2 1 1 1 1.

Even more unexpected is that after June 30, 2014, the situation changed abruptly (literally overnight) in the opposite direction − almost all clusterings became similar to the latter type (which may be called "degenerate"). There was no apparent market crisis at that time. In essence, one could say that, in these cases, no meaningful clusters exist at all. Recall that clusters are determined by the similar price behavior of the respective stocks.

It appears that the key lies in a sudden shift in the ratio between the frequency of data collection and the rate of change of stock prices. Evidently, on that specific day, the operational speed of the main programs governing the stock market was altered. This implies that the number of transactions per unit time (for instance, per day) increased dramatically (by a factor of ten or more), whereas the stock price data used in this paper remained available only once per day (at market closing). Consequently, the resulting clusterings become practically meaningless. Note that the situation would not improve by employing various methods to avoid small clusters; in the absence of a genuine cluster structure, no reliable results can be obtained regardless.

What to do in this situation? One could, for example, use autocorrelation coefficients to estimate the rate at which prices change, then select an appropriate frequency for data collection so that the ratio of these frequencies (price changes versus data collection) remains roughly the same as it is now. There is no reason to believe that market behavior on shorter intervals would be fundamentally different. One may hope that, as in many other scenarios, a kind of **fractality** emerges − here not as a similarity in order, but as a similarity in disorder.

Of course, achieving this would require substantial research well beyond the scope of the present article, yet still grounded in the approaches outlined herein.

2. The sums reported in Tables 4 and 5 may seem insignificant in the context of the multibillion-dollar S&P-500 stock market. However, recall that QTA considers no more than five sets of promising stocks, and in each set, only one share of each type is traded. If one were to buy and sell 100 shares instead of just one, the returns would be multiplied by 100, and so on. Furthermore, the different trading variants run simultaneously, and the total profit equals the sum of the profits accumulated across each variant (see Table 5). Their number can be substantially increased.

3. The idea of switching trading to a backward mode was inspired by a well-known scheme in optimal control theory: bringing a rocket to zero velocity by alternating forward and backward thrust. In the algorithm under consideration, such a switch can occur at most once. The results might be improved by considering trading not on a quarterly basis but over longer intervals (e.g., years). In that case, one could employ multiple mode switches, depending on the evolving situation − one that may not fully play out within a single quarter.

4. A careful reader may note the following. When trading opens on day t, the chosen stocks prices can differ slightly from their closing prices on the previous day. Yet the profit expression involves differences between closing prices on days $t-1$ and t. As a result, the gains realized may differ marginally from those computed by the proposed algorithm.

The correction becomes straightforward once stock prices are made available more frequently (see item 1 above in Conclusion). The algorithms themselves require no modification.

The authors extend their gratitude to Professor F. T. Aleskerov for his support of this work and to Professor B. G. Mirkin for his assistance in preparing this article.

References

1. Jallo D., Budai D., Boginski V., Goldengorin B., Pardalos P. M. (2013). Network-Based Representation of Stock Market Dynamics: An Application to American and Swedish Stock Markets. Springer Proceedings in Mathematics & Statistics, Vol. 32, 91–108.

2. Goldengorin B., Kocheturov A., Pardalos P.M. (2014). A Pseudo-Boolean Approach to the Market Graph Analysis by Means of the p-Median Model. In Honor of Boris Mirkin's 70th Birthday: Clusters, Orders, and

Trees: Methods and Applications. Springer Optimization and Its Applications, Vol. 92, 77–89.
3. Lohrmann C., Luukka P. (2019) Classification of intraday S&P500 returns with a Random Forest. International Journal of Forecasting. Vol. 35, Issue 1, 28–39.
4. Yanru Guo. (2020) Stock Trading Based on Principal Component Analysis and Clustering Analysis. IOP Conf. Series: Materials Science and Engineering, 740–748.
5. Bruni R. (2017) Stock Market Index Data and indicators for Day Trading as a Binary Classification problem. Data in Brief, Vol. 10, 569–575
6. P.Y. Fung. (2021) Online two-way trading: Randomization and advice. Theoretical Computer Science. Vol. 856, 8, 41–50
7. Rubchinsky A. Divisive-Agglomerative Classification Algorithm Based on the Minimax Modification of Frequency Approach: Preprint / NRU HSE / WP7/2010/07. M.: 2010. 48 p. //Available in Internet at the address: www.hse.ru/en/org/hse/wp/wp7en
8. Newman, M.E.J., Girvan, M. Community structure in social and biological networks. – Proc. Natl. Acad. Sci. USA 99, 7821–7826, 2002
9. Rubchinsky, A., Baikova, K. (2023). Algorithm of Trading on the Stock Market, Providing Satisfactory Results. In: Goldengorin, B., Kuznetsov, S. (eds) Data Analysis and Optimization. Springer Optimization and Its Applications, vol 202. Springer, Cham, 331–347

Рубчинский А. А, Чубарова Д. А.

Эффективный алгоритм торговли на фондовом рынке: ретроспективный анализ, основанный на данных по S&P-500 [Электронный ресурс]: препринт WP7/2025/01 / А. А. Рубчинский, Д. А. Чубарова; Нац. исслед. ун-т «Высшая школа экономики». – Электрон. текст. дан. (900 Кб). – М.: Изд. дом Высшей школы экономики, 2025. – (Серия WP7 «Математические методы анализа решений в экономике, бизнесе и политике»). – 37 с. – На англ. яз.

В статье рассматривается один из наиболее известных примеров социаально-экономических систем, характеризующихся значительной неопределенностью – фондовый рынок S&P-500, на котором торгуются акции 500 крупнейших компаний США. Никаких предположений о вероятностных характеристиках фондового рынка не делается. Был разработан гибкий алгоритм ежедневной торговли, основанный как на известных фиксированных данных о стоимости акций за предыдущие дни, так и на доходах и расходах, уже полученных с начала текущего квартала. Результаты вычислительных экспериментов демонстрируют устойчивый рост накопленного дохода в 1995–2014 годах, а также возможность использования докризисного и посткризисного периодов для надежного увеличения доходов от торговли акциями.

.

Препринты Национального исследовательского университета
≪Высшая школа экономики≫ размещаются по адресу: http://www.hse.ru/org/hse/wp

Препринт WP7/2025/01
Серия WP7
Математические методы анализа решений
в экономике, бизнесе и политике

Рубчинский Александр Анатольевич,
Чубарова Дарья Алексеевна

Эффективный алгоритм торговли на фондовом рынке: ретроспективный анализ, основанный на данных по S&P-500

(*на английском языке*)

Публикуется в авторской редакции

Изд. № 2965