HIGHER SCHOOL OF ECONOMICS

NATIONAL RESEARCH UNIVERSITY

Alexander Rubchinsky

FAMILY OF GRAPH DECOMPOSITIONS AND ITS APPLICATIONS TO DATA ANALYSIS

Working Paper WP7/2016/09
Series WP7
Mathematical methods
for decision making in economics,
business and politics

Moscow
2016

УДК 519.8
ББК 22.176
R88

Editors of the Series WP7
"Mathematical methods for decision making
in economics, business and politics"
Aleskerov Fuad, Mirkin Boris, Podinovskiy Vladislav

Rubchinsky, A.
R88 Family of Graph Decompositions and Its Applications to Data Analysis [Text] : Working paper WP7/2016/09 / A. Rubchinsky ; National Research University Higher School of Economics. – Moscow : Higher School of Economics Publ. House, 2016. – (Series WP7 "Mathematical methods for decision making in economics, business and politics"). – 60 p. – 35 copies.

A new decomposition approach to complex systems analysis is suggested. The conventional approach deals with the construction of a single, "the most correct", decomposition of the considered system. Meanwhile the suggested approach is oriented to the construction of a family of decompositions, whose properties reveal some important meaningful features of the initial system.

The expedience and applicability of the elaborated approach are illustrated by three well-known and important cases: automatic classification, political voting body and stock market. In these cases, the presented results cannot be obtained by other known methods. These examples confirm the advantages of the suggested approach.

Key words: graph decomposition; automatic classification; complexity index; entropy; voting political body; ctock market; crisis

УДК 519.8
ББК 22.176

Rubchinsky Alexander, National Research University "Higher School of Economics"; National Research Technological University "MISIS"; arubchinsky@yahoo.com

© Alexander Rubchinsky, 2016
© National Research University
Higher School of Economics, 2016

1. Introduction

The graph is one of the most convenient and widespread models of many real and formal systems. Therefore, it is not surprising that the problem of graph decomposition is one of the very important and comprehensively studied problems in graph theory, computer science, data mining and other fields of discrete applied mathematics. There are hundreds of decomposition problem statements, concerning decomposition into subgraphs of a prespecified type, into paths, cycles, into product of other graphs, and so on. Yet we refer the subject to the most naïve and simple form: divide an initial undirected graph into two or more subgraphs, so that the number of edges inside these subgraps significally exceeds the number of edges, connecting different subgraps. Of course, this statement is not a formal one. Some formal refinements of this informal statement will be considered below, in subsection 2.2.

The desirable decomposition of a graph into a small number of clearly distinct (i.e. loosely connected) subgraphs can, in its turn, lead to important conclusions concerning the initial system, modeling by the graph. It allows formulating reasonable hypotheses about the system behavior, selecting a few important parameters, and so on − in brief, allows understanding «what is the world in this location».

However, in some cases, such a decomposition does not exist or there are several different decompositions of such type. These cases are not vexatious mistakes. Moreover, it is possible to assert that numerous decompositions naturally arise in the study of many complex systems, including systems, whose functioning is determined by human activity. Hence, it seems expedient to consider − as an important generalization of the conventional decomposition problem – construction of a ***family of decompositions*** instead of a single one.

As in the conventional case, the constructed family of decompositions characterizes the initial system. Moreover, in such situations decomposetions ***themselves***, ***forming the above-mentioned family***, ***are of little interest***. It turned out that it is much more expedient to focus our attention on calculation of special numerical indices based on these families. The suggested indices describe such properties of the initial system that cannot be revealed by the conventional decomposition approach. They have different meaningful interpretations in different situations, but generally they describe complexity, entanglement, intricacy and other similar hardly defined, though important, properties of various real systems. Therefore, these indices are referred to as ***decomposition complexity indices***. It should be mentioned that ***decomposition complexity*** concept introduced here is slightly akin to various concepts of system's complexity (see, for instance, [Pincus, 1991; McCabe, 1976; Raychev, 2016]), although it concerns much deeper levels of the structural organization.

The suggested approach to graph decompositions turns out to be helpful in data analysis, especially in order to extract some hidden facts from raw data. Three different applications of the approach are considered below.

The presented material is structured as follows.

1. Introduction.
2. Family of decompositions construction.
3. Formal definition of decomposition complexity indices of a given graph.
4. Applications for automatic classification.
5. Applications for analysis of voting political bodies.
6. Applications for analysis of crises in stock markets.
7. Conclusion.

Materials, related to sections 2 – 5, were partly presented in the recent working papers [Rubchinsky, 2015, 2010]. Material from section 6 is completely the new one. It concerns very important, difficult and unsolved problem of short-term prediction of crises in stock markets, based on share prices at some period (several days) prior to the crisis. In spite of limited and incomplete character of obtained results, it seems that they present the first step in the right direction.

2. Construction of Family of Graph Decompositions

The graph decomposition algorithm, presented below, firstly was suggested in [Rubchinsky, 2010] as an automatic classification algorithm. It was ***aimed at constructing of a single classification***, like other classification algorithms, which had been elaborated by this moment. However, just the same algorithm ***can be successfully used for solution of important and difficult real-life problems***, ***whose statements in themselves do***

not concern any classification. The matter is that the above-mentioned algorithm produces a ***family of classifications at every run***. As a conventional classification algorithm, it selects a single classification – namely, the classification with the maximum number of classes, such that at every algorithm run belongs to the family, produced at this run. Such a classification in many cases does not exist, but this algorithm can be used in framework of the suggested new approach – as the algorithm, producing a family of graph decompositions.

Current Section 2 is devoted to formal description of the algorithm of construction of a family of decompositions. The input of the algorithm is an undirected graph. The algorithm is determined as a three-level procedure. Several examples, illustrating the introduced notions, are presented in this Section 2.

The ***Algorithm of Graph Dichotomy***, which was firstly described in [Rubchinsky, 2010] presents the ***internal level*** of the general three-level procedure (see subsection 2.1).

The ***Divisive-Agglomerative Algorithm*** (DAA for brevity), which also was firstly described in [Rubchinsky, 2010], is based on the above-mentioned algorithm of graph dichotomy. This algorithm forms the ***intermediate level*** of the general procedure. It produces one family of decomposetions (see subsection 2.2). Note that some decomposition of the constructed family can coincide with one another.

At the ***external level*** several runs of DAA are accomplished. Every such a run determines a family of decompositions. The union (over all these runs) of all the constructed families forms a family of decomposetions (see subsection 2.3). Unlike internal and intermediate levels, the new version of operations at this level is firstly described here.

Let us consider the above-mentioned levels in more detail.

2.1. Internal level − Algorithm of the Graph Dichotomy

Let us start with an historical journey. In the article "Community structure in social and biological networks" [Girvan, Newman, 2002] a new approach to graphs decomposition was suggested. Let us describe the essence of the matter, citing the article.

"We define the ***edge betweenness*** of an edge as the number of shortest paths between pairs of vertices that run along it. If there is more than one shortest path between a pair of vertices, each path is given equal weight such that the total weight of all the paths is unity. If a network contains communities or groups that are only loosely connected by a few inter-group edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness. By removing these edges, we separate groups from one another and so reveal the underlying community structure of the graph."

Instead of edge betweenness, it is better to name the analogous notion as edge frequency. In slightly differing form, the Girvan-Newton algorithm is stated in [Rubchinsky, 2010] as follows.

Modified Girvan-Newman Algorithm

1. *Initial Setting*. Set the current frequency at every edge equal to zero.
2. *Vertices Choice*. Randomly choose two different vertices of the graph.
3. *Path Construction and Stop Checking*. Find a path between the two vertices. If such a path does not exist, go to step 7.
4. *Frequencies Update*. Add one to frequencies in all the edges on the path found at step 3.
5. *Continuation Checking*. If the number of consecutive runs of steps 2 – 4 does not exceed a pre-specified number M, go to step 2. Otherwise, go to the next step 6.
6. *Edge Removing*. Remove an edge with the maximum frequency and return to step 1.
7. *Stop*. Graph G is divided into two or more connected components, which are output as the final classes.

It is clear that during the execution of the algorithm every increment (by 1) of the number of connected components means division of one of groups into two parts, that is an hierarchical structure of groups (or communities) determined only by the initial graph, is obtained as a result.

In the above-described version of Girvan-Newman algorithm, there are some new features:

- use of random paths (instead of shortest ones) for calculation of edges betweenness;
- use of relatively small part of the set of all pairs of vertices (instead of all of them) for estimation of edge betweenness;
- edge removal based on this estimation.

The obvious drawback of Girvan-Newman algorithm (outlined by its authors) is that after removal of an edge with the highest frequency at step 6 all the accumulated statistics about edge frequency are deleted, and, hence, the statistics are not used subsequently. If it was possible to save these data for consecutive steps, it could essentially accelerate the algorithm. About this issue in the already cited article [Girvan, Newman, 2002] the authors wrote the following. "To try to reduce the running time of the algorithm further, one might be tempted to calculate the betweennesses of all edges only once and then remove them in order of decreesing betweenness. We find however that this strategy does not work well, because if two communities are connected by more than one edge, then there is no guarantee that all of those edges will have high betweenness – we only know that at least one of them will. By recalculating betweennesses after the removal of each edge we ensure that at least one of the remaining edges between two communities will always have a high value."

However, the dichotomy algorithm, described further on in the current subsection 2.1, avoids this trap. The essence of the matter is as follows.

In the previously considered frequency algorithms, paths, connecting a next pair of vertices, were traced ***independently of all the already traced paths***. Yet, taking into account all the already traced paths can obtain cuts between two sets of vertices whose all the edges have the ***same maximum frequency***. Then concurrent removal of all the edges with the maximum frequency defines the desired dichotomy of the graph.

Frequency algorithm of the graph dichotomy. The input of the algorithm is an undirected graph *G*. There are two integer parameters of the algorithm:

– the maximum initial value *f* of edge frequency;

– the number of repetitions *T* for statistical estimation of edges frequency.

1. *Initial setting*.

1.1. Finding connected components of the given graph *G* (by any standard algorithm).

1.2. If the number of components is greater than 1 (i.e. graph *G* is disconnected), then the component with the maximum number of vertices is declared as the first part of the constructed dichotomy of the initial graph;

all the other components form its second part; thus, the ***dichotomy is constructed and the algorithm stops***. Otherwise, go to the next step 1.3.

1.3. Integer numbers uniformly distributed on the segment $[0, f-1]$ initialize frequencies in all the edges. This operation accelerates convergence of frequencies.

2. *Cumulative stage.* All the operations of this stage are repeated T times:

2.1. Random choice of a pair of different vertices of graph G.

2.2. Construction of a minimal path (connecting the two chosen vertices so that its longest edge is the shortest one among all such paths) by Dijkstra algorithm (see, for instance, [Goodman, Hedetniemi, 1977]). The length of an edge is its current frequency.

2.3. Frequencies modification. The value 1 is added to the frequencies in all the edges on the path found at the previous step 2.2.

3. *Final stage*.

3.1. The maximum value of frequency $f_{\max}$ over all the edges is stored.

3.2. The operations of steps 2.1 – 2.3 from cumulative stage are executed once.

3.3. The maximum value of frequency f_{mod} in edges is found.

3.4. If $f_{\text{mod}} = f_{\max}$, go to step 3.2; otherwise, go to the next step 3.5.

3.5. Deduct the value 1 from the frequencies in all the edges forming the last found path.

3.6. Remove all the edges, in which frequency is equal to $f_{\max}$.

3.7. Find connected components of the modified graph. The component with the maximum number of vertices is declared as the first part of the constructed dichotomy of the initial graph; all the other components form its second part. After that all the edges, removed at step 3.6, except the edges, connecting vertices from different parts of the dichotomy (if they exist) are returned into both subgraphs.

Let us give some comments to the described algorithm.

Comment 1. No action is taking if the initial graph is not the connected one (it has at least two connected components).

Comment 2. In order to construct no less than two subgraphs at step 3.7, we must be sure that the modified graph, constructed at step 3.6, is not the connected one (it has at least two connected components). The answer is given by the following simple reasoning.

Let us consider frequencies of all the edges just before the construction of a last path. Assume that at this moment all the edges with the maximum value of frequency $f_{\max}$ do not contain any cut of the graph (remember that graph G is the connected graph). Therefore, for any two vertices there are paths (at least one), so that frequencies in all the edges of this path are less than $f_{\max}$. Dijkstra algorithm will find one such a path, whose every edge has a frequency less than $f_{\max}$. Hence, after adding 1 to frequencies in all these edges, the modified value f_{mod} cannot be greater than $f_{\max}$. Following the comparison at step 3.4, we must go to step 3.2 and find another path. But we supposed that the already considered path was the last one. Therefore, edges with maximum frequency contain some cut and their elimination makes the graph disconnected.

Comment 3. Another important question concerns the same step 3.7. The result of operations at this step is a dichotomy, i.e. a division of the initial graph into two parts. Because the algorithm uses a random generator as well as two external parameters, how can we be sure that the constructed dichotomy is the same, independently of the parameters choice and random generator initialization? This question turns out to be the crucial in the framework of the suggested approach. It will be comprehensively discussed below in the material.

Example 1. Consider graph, shown in Fig. 1a. Let us apply to this graph the above algorithm. The accomplishment of the algorithm includes the consecutive random choice of 50 pairs of vertices (plus a few random pairs at the final stage). For every pair a shortest path is constructed in correspondence to step 2.2. Remember that the length of a path is the length of its longest edge and the length of an edge is its current frequency.

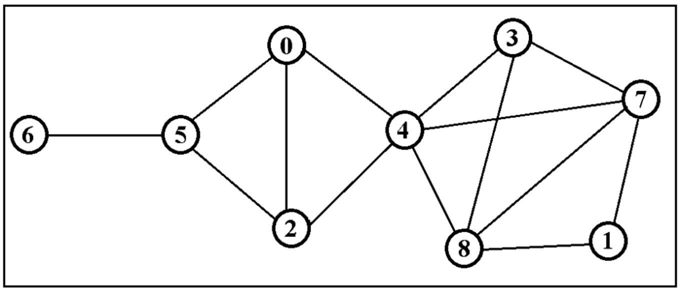

Fig. 1a. The initial graph

Fig. 1b shows the accumulated frequencies under parameters f = 5, T = 50. Fig. 1c and 1d show the accumulated frequencies under parameters f =5, T = 500. The last two cases differ only in initialization of random generator.

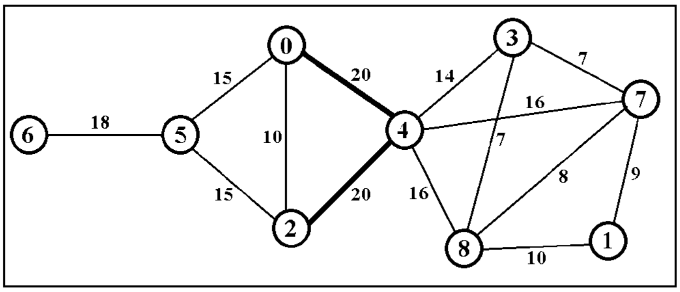

Fig. 1b. The accumulated frequencies under f = 5, T = 50

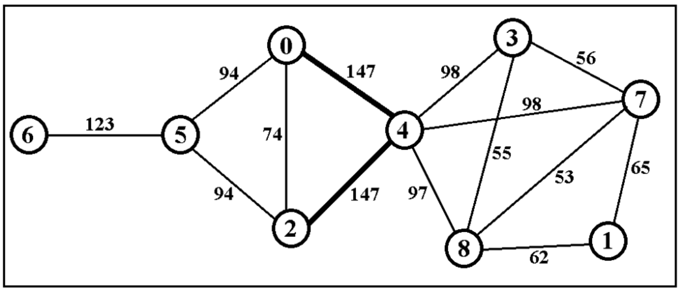

Fig. 1c. The accumulated frequencies under f = 5, T = 500 (case 1)

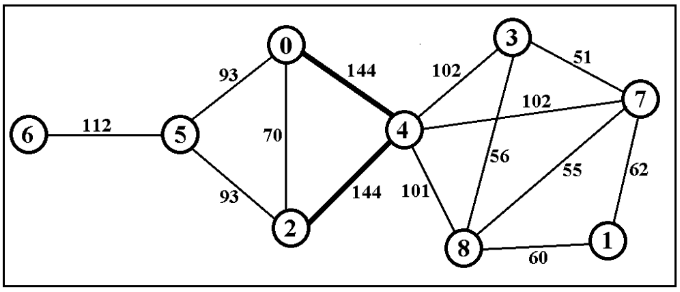

Fig. 1d. The accumulated frequencies under f = 5, T = 500 (case 2)

The sets of edges with a maximum value of frequency are the same in all the three cases. Bold lines in Fig. 1b, 1c and 1d show them. Thus, the dichotomies coincide in all the three cases (see Fig. 1e).

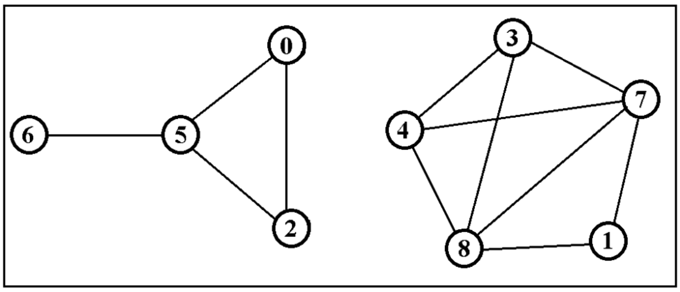

Fig. 1e. The single dichotomy

2.2. Intermediate Level – Divisive-Agglomerative Algorithm

This subsection is devoted to description of this algorithm. It consists of k consecutive steps, where k is the single parameter of the algorithm. The single input of the algorithm is the initial undirected graph G. Its output is a family of decompositions of the initial graph.

Divisive-Agglomerative Algorithm (DAA)

1. *Initial setting*. Assume $L_1^1 = G$, $p = 1$.
2. *Main step*. It includes two parts.

Part A. Construction of decomposition L_{p+1}^{p+1} into p+1 subgraphs, starting with decomposition L_p^p into p subgraphs (divisive stage).

Part B. Construction of family of decomposition L_p^{p+1}, …, L_2^{p+1} starting with one decomposition L_{p+1}^{p+1}, found in part A (agglomerative stage).

Description of Part A. Among all the p subgraphs of decomposition L_p^p select the subgraph with the maximum number of vertices. Divide it into two subgraphs by the above-described graph dichotomy algorithm (see subsection 2.1). Other $p-1$ subgraphs of L_p^p, together with just found two new subgraphs form new decomposition L_{p+1}^{p+1} into p+1 subgraphs.

Description of Part B. Among all the pairs of subgraphs, forming the decomposition L_{p+1}^{p+1}, find the pair, connected by the maximum number of edges, and pool them together. The obtained decomposition into p subgraphs is denoted by L_p^{p+1}. Analogously, by pooling together pairs of sub-graphs, connected by the maximum number of edges, we determine de-composition L_{p-1}^{p+1} from L_p^{p+1}, ..., and so on, till to the last decomposition L_2^{p+1} into two subgraphs.

3. *Stop condition checking*. If $p < k$, assume $p = p+1$ and go to Step 2.

4. *Output*. The family of found decompositions

$$L_2^2;\, L_2^3, L_3^3;\, L_2^4, L_3^4, L_4^4;\, \ldots;\, L_2^{k+1}, L_3^{k+1}, \ldots, L_{k+1}^{k+1}. \tag{1}$$

is the output of DAA.

It is easy to see that the number of decomposition in list (1) is equal to $\frac{(k+1)k}{2}$. Some of them can coincide, yet the minimum number of different decompositions is equal to k, because in the list (1) there are at least k different decompositions L_2^2, L_3^3, ..., L_k^k, L_{k+1}^{k+1} into 2, 3, …, k, k+1 subgraphs. These decompositions are referred to as the ***essential*** ones, whereas all the other decompositions are referred to as the ***adjoined*** ones. By the construction, every subgraph in every decomposition in the list (1) coincides with one of subgraphs from L_{k+1}^{k+1} or is the union of some of them.

Denote the family of decompositions, presented by list (1), as $L(k)$. It is evident that for any p ($1 \leq p < k$ the family of the first $\frac{(p+1)p}{2}$ decompositions from list (1) can be considered as $L(p)$.

The examples of accomplishment of DAA are presented below, in Section 4.

2.3. External Level – Repetitive Divisive-Agglomerative Algorithm Runs

At the external level, DAA is applied many times to the same initial graph. However, its output (list $L(k)$ of found decompositions) can differ in different runs. The matter is that at every step of the accumulating stage of the graph dichotomy algorithm (see subsection 2.1) a pair of vertices that must be connected by a path is selected randomly. It implies that output of DAA depends upon the initialization of random generator. More precisely, some decompositions at its different runs differ one to another, whereas some decompositions coincide at all the runs. Therefore, it is necessary to complete several runs of the same algorithm with the same initial data − otherwise, it is simply impossible to find out in one or another actual situation.

There are two parameters of the external level – counting numbers r and s. There are two loops of the runs – inner and outer. At the inner loop, DAA runs r times. It produces r families $L(k)$, consisting of $\frac{(k+1)k}{2}$ decompositions each. The outer loop consists of s runs of the above inner loop. Therefore, it produces a family, consisting of $s \times r \times \frac{(k+1)k}{2}$ decompositions. Denote family of decompositions, constructed on i^{th} iteration of the outer loop and on j^{th} iteration of the inner loop as $F(k, i, j)$ ($i = 1, \ldots, s$; $j = 1, \ldots, r$). Assume

$$F(k, i, r) = \cup_{j=1}^{r} F(k, i, j) \ (i = 1, \ldots, s), \tag{2}$$

$$F(k, r, s) = \cup_{i=1}^{s} F(k, i, r). \tag{3}$$

The latest big family $F(k, r, s)$ of decompositions is the output of the external level. It depends on parameters r and s. This family is declared as the output of the entire process. Necessary explanations about the described two-dimensional structure of the constructed family and its dependence on parameters k, r, s are presented in the next Section 3.

3. Indices of Decomposition Complexity of Graph

At the previous subsection 2.3, the two-dimensional family $F(k, r, s)$ of decompositions was constructed. This family was presented as unions of families $F(k, i, j)$. Every family $F(k, i, j)$ consists of k decompositions into 2 classes, $k-1$ decompositions into 3 classes, and so on – till to one decomposition into k classes (see subsection 2.2). Some of them can coin-cide and some of them can be different.

DAA includes multiple random operations at every run of graph dichotomy algorithm (see subsection 2.1). Therefore, its output – a family of $\frac{(k+1)k}{2}$ decompositions – also is of random character. Numbers, determined by these families, can be considered as ***random values***. The average values of these random numbers are determined as the above-mentioned indices, expressing important properties of considered systems.

3.1. Formal Definition of Indices of Decomposition Complexity

Let us introduce the necessary formal constructions and definitions.

Index 1. Denote

$$Q(k, i, r) = d/M \ (i = 1, \ldots, s), \tag{4}$$

where M is equal to $r\times\frac{(k+1)k}{2}$ (general number of decompositions in family $F(k, i, r)$), d is equal to the number of ***different*** decompositions among all the decompositions in the family $F(k, i, r)$.

In the most important cases numbers $Q(k, i, r)$ do not have a limit as functions of r when r tends to infinity, while k and i are arbitrary fixed values. Therefore, in order to find some stable answer it is suggested to average the numbers $Q(k, i, r)$ over parameter s – the length of the outer loop – as follows:

$$Q(k, r, s) = \frac{1}{s}\times\sum_{i=1}^{s} Q(k, i, r). \tag{5}$$

In opposite to values $Q(k, i, r)$, values $Q(k, r, s)$ converge than s tends to infinity (for arbitrary fixed k and r). Thus, $Q(k, r, s)$ is the average of values $Q(k, i, r)$ (over s) that are calculated separately for each family $F(k, i, r)$) ($i = 1, \ldots, s$). By the construction (see formula (4)),

$$0 < Q(k, i, r) \leq 1 \ (i = 1, \ldots, s)$$

that implies the analogous inequality for value $Q(k, r, s)$. This value is defined as ***index* 1 *of decomposition complexity of the initial graph***. Index $Q(k, r, s)$ depends on parameters k, r, s; however, the experiments confirm that it tends to a finite limit when s tends to infinity. Really, values $Q(k, r, s)$ are very close to the limit then s exceeds 10.

Index 2. It is defined similarly index 1. Consider family $F(k, i, r)$ (i = 1, …, s). Assume that in this family a classification c_p encounters m_p times ($p = 1, \ldots, t$), where $\sum_{p=1}^{t} m_p = M$ (remember that $M = r\times\frac{(k+1)k}{2}$). Let us denote

$$E(k, i, r) = -\sum_{p=1}^{t} \mu_p \ln\left(\mu_p\right), \text{ where } \mu_p = m_p/M \ (i = 1, \ldots, s). \tag{6}$$

$E(k, i, r)$ is the conventional entropy of division of finite family $F(k, i, r)$ into subsets, consisting of coinciding classifications.

In the most important cases numbers $E(k, i, r)$ do not have a limit as functions of r when r tends to infinity, while k and i are arbitrary fixed values. Therefore, in order to find some stable answer it is suggested to average the numbers $E(k, i, r)$ over parameter s – the length of the outer loop – as follows:

$$E(k, r, s) = \frac{1}{s} \times \sum_{i=1}^{s} E(k, i, r). \quad (7)$$

In opposite to values $E(k, i, r)$, values $E(k, r, s)$ converge than s tends to infinity (for arbitrary fixed k and r). Thus, $E(k, r, s)$ is the average of values $E(k, i, r)$ (over s) that are calculated separately for each family $F(k, i, r)$ ($i = 1, \ldots, s$). By the construction (see formula (6)),

$0 \leq E(k, i, r) < \ln(M)$ ($i = 1, \ldots, s$),

that implies the analogous inequality for value $E(k, r, s)$. This value is defined as ***index* 2 *of decomposition complexity of the initial graph***. Index $E(k, r, s)$ depends on parameters k, r, s; however, the experiments confirm that it tends to a finite limit when s tends to infinity. Really, values $E(k, r, s)$ are very close to the limit then s exceeds 10.

Remember that values $Q(k, i, r)$ and $E(k, i, r)$ do not have a limit when r tends to infinity and k, i are fixed numbers.

Both indices describe complexity, intricacy, entanglement, and other hardly defined but important properties of an initial graph. It is intuitively clear that small (close to 0) values $Q(k, r, s)$ and $E(k, r, s)$ correspond to relatively simple situations. In these situations only k decompositions, obtained by successive divisions of the initial sets into 2, 3, …, k+1 subgraphs are different. Unions in agglomerative stages do not add new decompositions. Larger (close to the maximum possible) values $Q(k, r, s)$ and $E(k, r, s)$ correspond to relatively complex situations, in which found decompositions essentially depend upon random generator initialization, and adjoined decompositions differ from essential ones (see subsection 2.2).

Another time pay attention to formulae (5) and (7), determining considered decomposition complexity indices. The members of sum in the right-hand parts in both formulae do not converge, while both sums in the left-hand parts in these formulae do converge. These circumstances justify the suggested approach to decomposition complexity definition.

In this connection it is possible to remember (as an analog) example of random walk, thoroughly considered in chapter III of the famous book [Feller, 1969]. There the following game was studied. If an unbiased coin falls (after tossing) heads up, player B pays to player A \$1; otherwise, player A pays to player B \$1.

Denote the gain of player A for first k such games as Z_k. Of course, Z_k can be positive, negative or 0 (only for even numbers k). Denote by g_n^+ the number of values k between 0 and n, such that $Z_k \geq 0$, by g_n^- − the number of values k between 0 and n, such that $Z_k \leq 0$. Intuitively it seems that $\lim_{n\to\infty} g_n^+ / g_n^- = 1$. Yet in this book, it is proved that, in this case, intuition is wrong, and the above mentioned limit simply does not exist. It seems that it contradicts to the symmetry of the game. Nevertheless, the symmetry (existence of the limit equal to 1) is rebuilt if one considers simultaneous accomplishing of a large enough number of such games. It is possible to say that ***one arbitrary long sequence of single games*** does not converge, while the set of a ***large enough number of such sequences*** demonstrates convergence.

Of course, in the considered in the present work case mathematical essence of the absence of convergence in separate sequence is more complicated. The original cause consists in the suggested algorithm of graph dichotomy. It can produce significantly different divisions into two subgraphs under arbitrary number of random paths. It is not a mistake but the kernel of the suggested approach that constructed families of subgraphs generally differing one to another.

Even the existing limits of $Q(k, r, s)$ and $E(k, r, s)$ (when s tends to infinity) still depend on k and r. Theoretically important question about exact definitions of indices that do not depend upon all the parameters but only on the initial graph remains open.

3.2. Indices of Decomposition Complexity of Some Graphs

Because the considered indices are calculated basing only on the initial graph, they can be related to the arbitrary undirected graphs themselves, out of any meaningful connections to decompositions. Let us give some explanati-ons, concerning the introduced in current Section 3 both decomposition indices for some graphs.

Let us try to answer the following question – which graphs have low or high level of indices of decomposition complexity? For simplicity assume $k = 1$. It means that we will consider only dichotomies of graphs.

Assume $G(V, E)$ be any undirected graph with set of vertices V and set of edges E. It is supposed that $G(V, E)$ is a connected graph. For any subset $A \subset V$ denote $\overline{A} = V \setminus A$. Denote the number of edges, connecting sets A and $\overline{A}$ as $d(A, \overline{A})$, the number of elements in any finite set X as $|X|$. The expression

$$D(A) = \frac{|A| \times |\overline{A}|}{d(A, \overline{A})} \tag{8}$$

was named in [Rubchinsky, 2015] as ***decomposition function of graph***. This function is determined for all proper subsets A of V.

Let us consider the maximization problem

$$D(A) \to \max \tag{9}$$

over the finite set of all the proper subsets of V. It was shown in [Rubchinsky, 2015] that the suggested algorithm of graph dichotomy construct a cut $(A^*, \overline{A}^*)$, such that set A^* gives an approximate solution of the above-mentioned maximization problem.

It is easy to see that problem (9) is equivalent to the minimization problem

$$R(A) = d(A, \overline{A}\ ) \times (\frac{1}{|A|} + \frac{1}{|\overline{A}|}) \to \min, \tag{10}$$

well known as "Ratio Cut Problem" (see, for instance, [Luxburg, 2007]).

Let us introduce the necessary notions. Sets P and Q are close one to another if their symmetrical difference consists of few elements relatively to the lesser of these sets P and Q. All the sets, close to the same set X, form the neighborhood of X. The solution A^* of problem (9) is called the ***isolated*** one, if it is the only global maximum in the problem and for any sets A out of some neighborhood of A^* the value $D(A)$ is essentially less than $D(A^*)$.

Of course, the above definition is not the exact one. It simply gives some meaningful description of the considered situation. It is possible to complete this description with the following remark. Let us consider a graph with the following structure. This graph can be divided into two subgraphs, so that:

– the number of edges, connecting these subgraphs, is significant less than numbers of edges inside each of the two subgraphs,

– the ratio between the number of vertices in the greater subgraph and the number of vertices in the lesser subgraph does not exceed 5-6.

The computational experiments demonstrate that in such cases the algorithm of graph dichotomy finds only one division independently of any initialization of the random generator. Therefore, values $Q(1, r, s)$ and $E(1, r, s)$ are close to 0 for arbitrary r and s. We do not discuss here, how to establish the existence of such a structure.

Let us consider simple examples of graph with special (large and small) values of the both decomposition complexity indices.

Example 2. Let us consider the complete graph with any number of vertices. If a subset of vertices maximizes the decomposition function D, then any other subset, containing the same number of vertices, will maximizes this function, too, because of the symmetrical situation. Therefore, every run of the graph dichotomy algorithm produces a random dichotomy, typically consisting of two subgraphs with almost equal numbers of vertices. In such cases both indices are close to the maximum possible values (1 for index Q and $\ln(M)$ for index E).

Example 3. Let us consider the simple graph, consisting of one cycle with N vertices. This graph will be divided into two parts, containing approximately $N/2$ vertices each. Any such halve is a chain of adjoined vertices. From the symmetry, all such chains are equiprobable. Therefore, both indices are close to the maximum possible values (1 for index Q and $\ln(M)$ for index E), like in the previously considered graph from Example 2.

Example 4. Let us consider the graph from Example 1, shown in Fig. 1a. Both indices have the minimum values 0 and 1, because all the decompositions into 2 classes coincide.

Comment 4. From any intuitively reasonable and even from the most formal points of view the complexity of the first of three just mentioned graphs is the maximum one, the complexity of the second of three just mentioned graphs is the minimum one, and the complexity of the third of three just mentioned graphs is intermediate one. However, the suggested here measures of complexity (indices of decomposition complexity) are of another kind. The exact analysis of these indices allows us reaching to non-evident but important conclusions in some real situations.

4. Applications in Automatic Classification

The well-known automatic classification problem (further referred for brevity as AC) consists in the division of a given set of objects into several non-intersecting subsets (usually called classes, aggregates, clusters, etc.). It is required that objects belonging to a same class are in some sense closely connected, similar in appearance, while objects belonging to different classes are distinct, as unlike as possible. Informal character of AC problem, its various statements and applications, numerous approaches and methods of its solution are comprehensively described in several monographs and reviews (see, for instance, [Mirkin, 2012 and 2011]).

The suggested here approach to AC problem consists of three main stages.

1. The initial data about the AC problem is presented by an undirected graph, so that any class consists of vertices, closely connected by many edges, while different classes are loosely connected by edges.

2. The family of the graph decompositions is constructed by the algorithm, described in Section 2.

3. The solution of the initial AC problem defined as the decomposition

with the maximum number of classes, which belongs to the constructed families at every DAA run. Such a classification in some cases does not exist and in some other cases it is not the unique one; however, these situations are not studied here.

Let us consider these stages separately, accompanying them by examples.

4.1. Reduction of AC Problem

Initial data in AC problems are mostly presented in one of the following form: a raw entity-to-feature data table, a dissimilarity matrix, and an undirected graph. The first one can be reduced to the second one; the second one can be reduced to the third one. Let us dwell on these reductions in more detail.

1. Reduction from raw entity-to-feature data table to dissimilarity matrix consists of two simple steps (see, for instance, [Mirkin, 2011]):

a) reduction from raw entity-to-feature data table to the standardized data table by the formula

$y_{ij} = \frac{x_{ij}-x_j^-}{x_j^+-x_j^-}$ (i = 1, ..., N; j = 1, ..., m),

where N is the number of entities, m is the number of features,

$x_j^- = \min_i x_{ij}$, $x_j^+ = \max_i x_{ij}$ (j = 1, ..., m);

b) reducing from standardized data table to dissimilarity matrix by the formula

$d_{st} = \sqrt{\sum_{i=1}^{m}(y_{si} - y_{ti})^2}$ (s = 1, ..., N–1; t = s+1, ..., N),

$d_{st} = d_{ts}$ ($s > t$), $d_{ss} = 0$ (s = 1, ..., N),

where y_{pq} are elements of standardized data table determined at step a).

2. Reduction from dissimilarity matrix to neighborhood graph. The notion of such a graph is well known (see, for instance, [Luxburg, 2007]). Graph vertices are in the one-to-one correspondence to the considered entities. For every entity (say, a) all the other vertices are ordered as follows: the distance between i^{th} object in the list and object a is a non-decreasing function of index i. The first four vertices in this list (i.e. the first four closest vertices) as well as all the other vertices (if they exist), whose distances from a are equal to the distance from a to the fourth vertex in the list, are connected by an edge to the vertex, corresponding to entity a. It is easy to see that the constructed graph does not depend on a specific numeration, satisfying the above conditions. The number of closest vertices is a parameter of the general algorithm of decompositions family construction. Here this parameter is assumed be equal to four.

3. Sometimes the initial system has been already presented as undirected graph. In such cases, we do not require any reduction.

Thus, we can present initial data of most AC problem in form of an undirected graph.

Example 5. Consider the set of points on the plane, shown in Fig. 2a. Element d_{ij} of the correspondent dissimilarity matrix D is equal to Euclidean distance between points i and j (i, j = 1, ..., N).

The neighborhood graph, corresponding to the set, shown in Fig. 2a, is presented in Fig. 2b. Despite the definition, prescribing to connect every vertex to four closest ones, degree of some vertices exceeds four. The explanation of this apparent contradiction is evident.

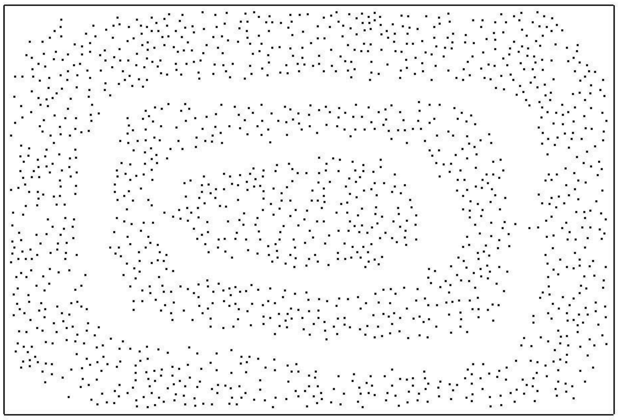

Fig. 2a. The initial set of points

4.2. Examples of AC Problems and Their Solutions

In this subsec-tion several examples of the work of the algorithm from Section 2 are considered.

Example 6. Consider the set of points on the plane, shown in Fig. 3a. The corresponding neighborhood graph is not shown, because the number of vertices is too large for reasonable graphic presentation.

Let us apply the algorithm of family of graph decomposition construction from Section 2 to the corresponding neighborhood graph. Assume k = 2 (see subsection 2.2), r = 3 and s = 1 (see subsection 2.3). After the three runs of the graph dichotomy algorithm we obtain three different essential decompositions L_2^2, shown in Fig. 3b – 3d. Only edges, forming the found cut, are shown in the next figures.

Following DAA, the graph dichotomy algorithm is applied to the greater subgraphs in every case. The different essential decompositions L_3^3 are shown in Fig. 3e – 3g. In the correspondence to the agglomerative stage of DAA, consider the found essential decompositions L_3^3 and pool together two subgraphs, connected by the maximum number of edges. In all the cases the resulting decomposition L_2^3 is the same. It is shown in Fig. 3h. Thus, only this decomposition is defined as the solution of the initial classification problem. Even is this simple situation practically all the classification methods fail (including popular spectral methods and balanced cut approach; see [Rubchinsky, 2010]).

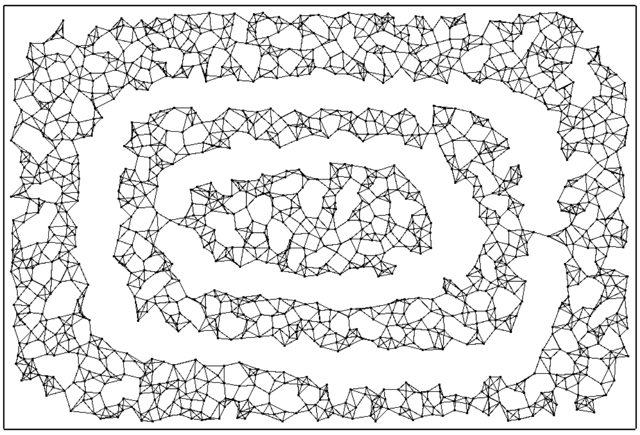

Fig. 2b. Neighborhood graph for the set in Fig. 2a

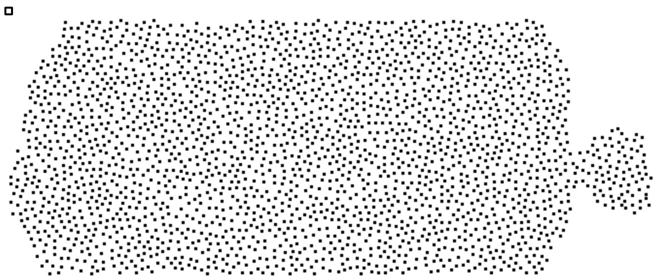

Fig. 3a. The initial set

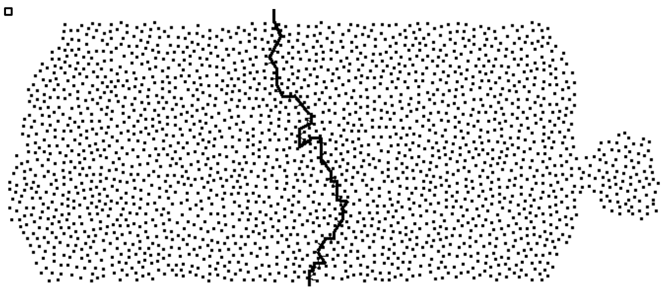

Fig. 3b. Dichotomy 1

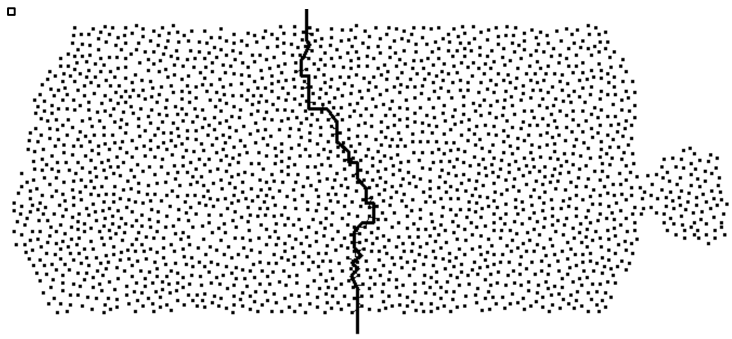

Fig.3c. Dichotomy 2

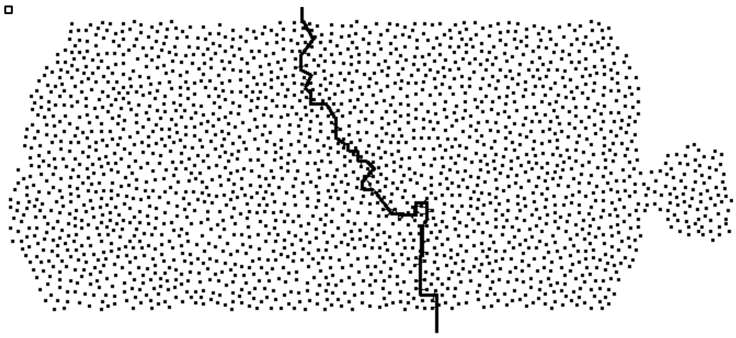

Fig. 3d. Dichotomy 3

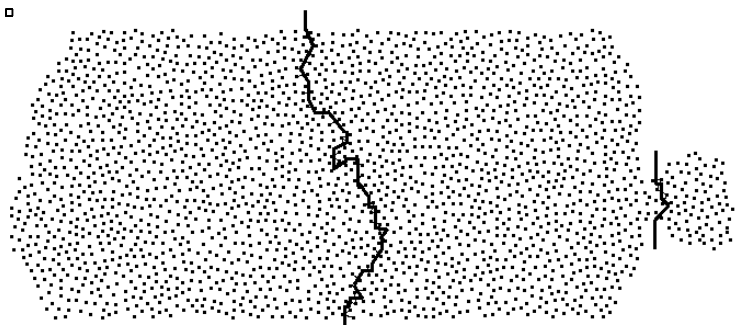

Fig. 3e. Decomposition L_3^3-1

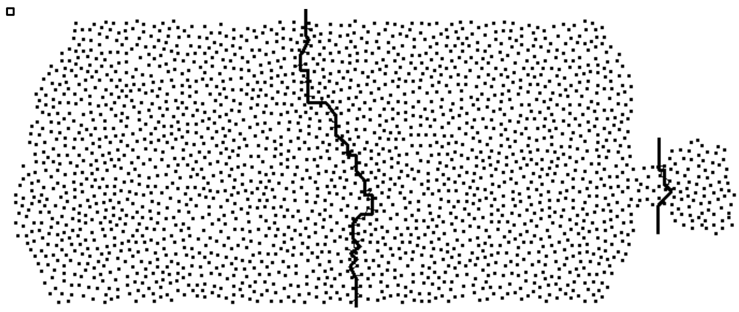

Fig. 3f. Decomposition L_3^3−2

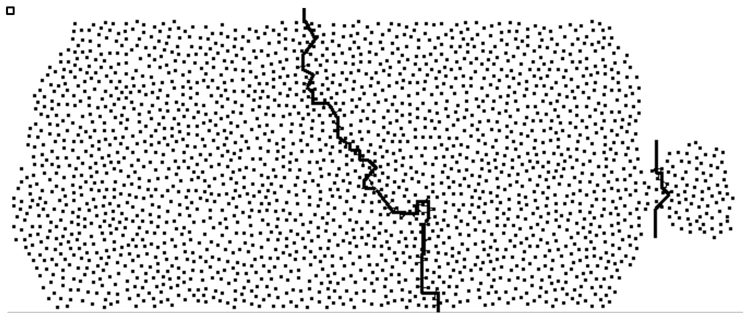

Fig. 3g. Decomposition L_3^3−3

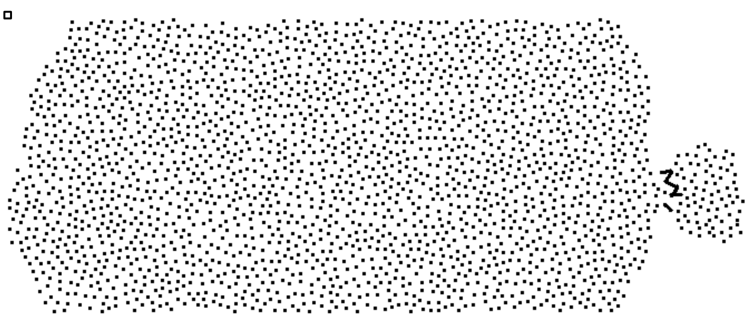

Fig. 3h. Decomposition L_2^3 − reasonable classification into two classes

Example 7. Let us demonstrate the application of the elaborated approach to AC problem in a more complicated case – the set of points, shown in Fig. 2a. Let us start with only one run of DAA. Consider consecutive dichotomies and construct essential and adjoined decompositions, using notations from subsection 2.2. Assume $k = 3$, i.e. restrict our consideration to 3 consecutive dichotomies. Essential decompositions L_2^2, L_3^3 and L_4^4 are shown in Fig. 4a – 4c. The edges forming cuts between different subgraphs are shown, too. Pooling in accordance to the algorithm subgraphs 0 and 2 from decomposition L_3^3 (connected by the maximum number of edges) results in adjoined decomposition L_2^3, coinciding with the essential decomposition L_2^2.

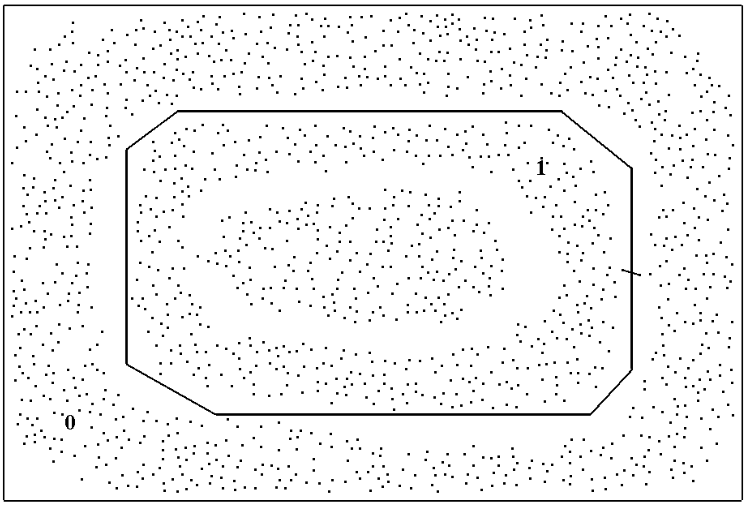

Fig. 4a. Essential decomposition L_2^2

After, pooling subgraphs 0 and 2 from decomposition L_4^4 shown in Fig. 4c, results in adjoined decomposition L_3^4, shown in Fig. 4d. It is clear that this classification is the only "correct" decomposition. However, DAA goal does not consists in finding one "correct" decomposition. It consists in the construction of family of decompositions for a given parameter k (number of consecutive dichotomies). Therefore, pooling subgraphs 0 and 1 from decomposition L_3^4 (that are connected by 2 edges), we obtain the adjoined decomposition L_2^4, which also coincides with essential decomposition L_2^2.

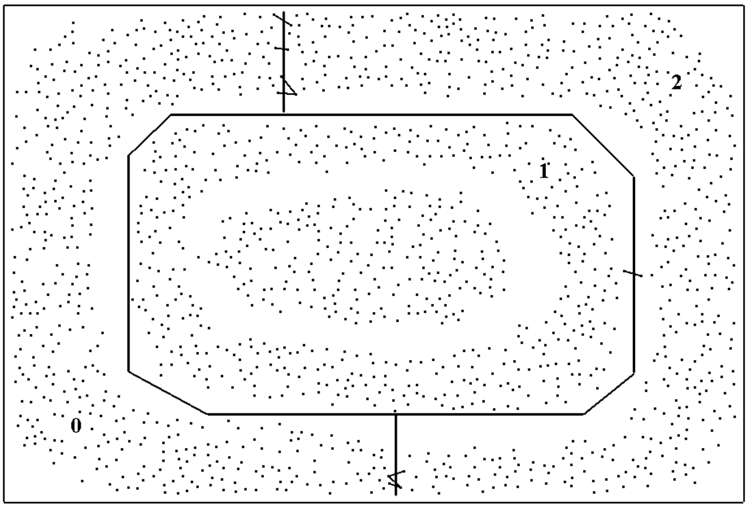

Fig. 4b. Essential decomposition L_3^3

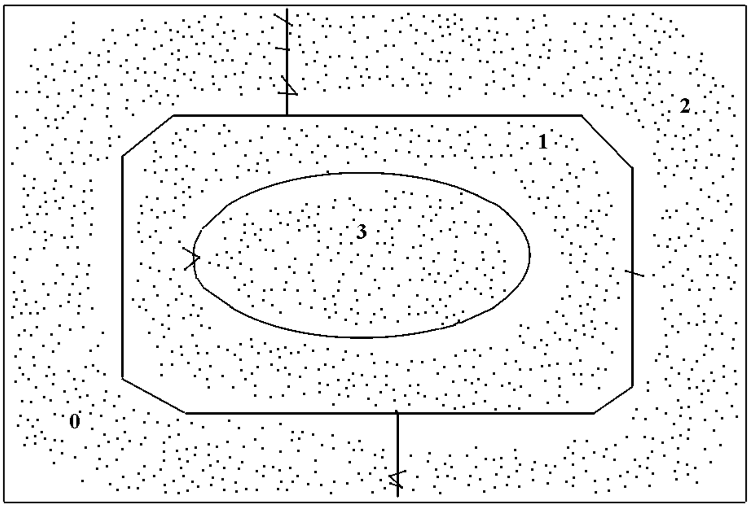

Fig. 4c. Essential decomposition L_4^4

At this point, one run of DAA is over. In every decomposition, found by this algorithm, every class is a union of classes, forming the last essential decomposition L_4^4 or coincides with one of them. Therefore, the scheme, shown in Fig. 5, naturally presents the process of its implementation in the considered case.

Finally, among all the possible 6 decompositions: L_2^2; L_2^3, L_3^3; L_2^4, L_3^4, L_4^4 from $L(3)$ there are 4 different decompositions: essential decompositions L_2^2, L_3^3 and L_4^4 and adjoined decomposition L_3^4 (see Fig 4a – 4d).

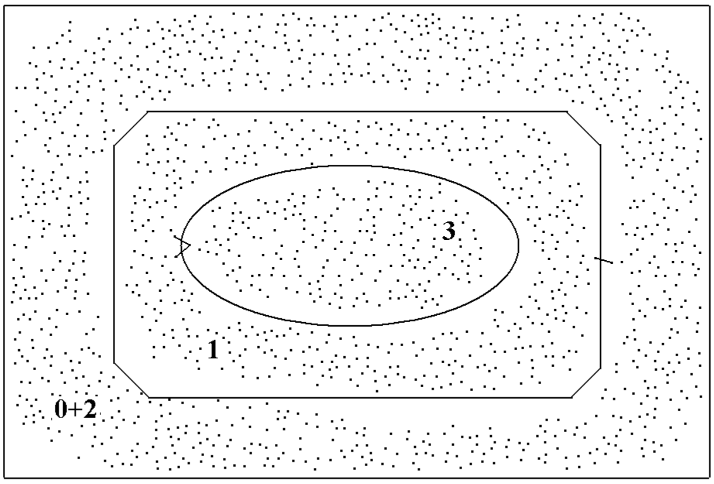

Fig. 4d. Adjoined decomposition L_3^4

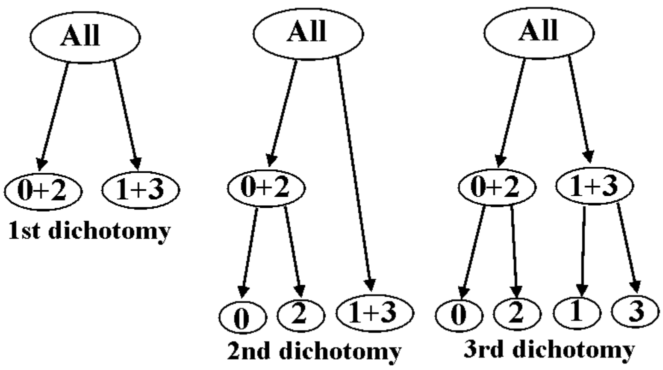

Fig. 5. DAA diagram

Assume (for visibility of illustration) the number r of DAA runs in the inner loop is equal to 4 and the number s of repetitions of the outer loop is equal to 1. In Fig. 6a – 6d, results of 4 runs for essential decomposition L_3^3 are shown (see also Fig. 4b). All the 4 found decompositions differ one from another.

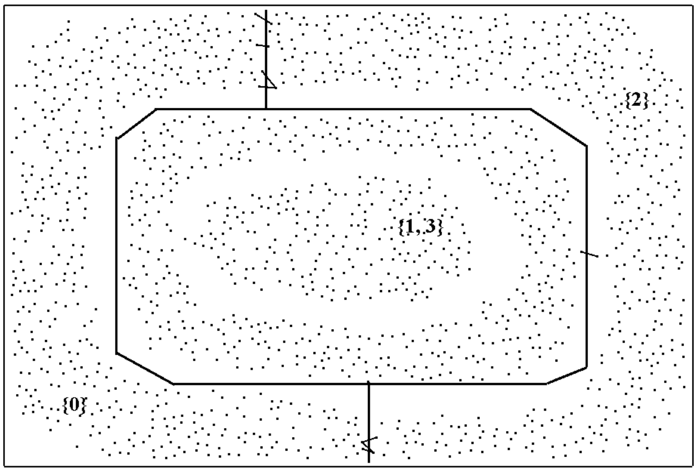

Fig. 6a. Decomposition L_3^3 found at run 1

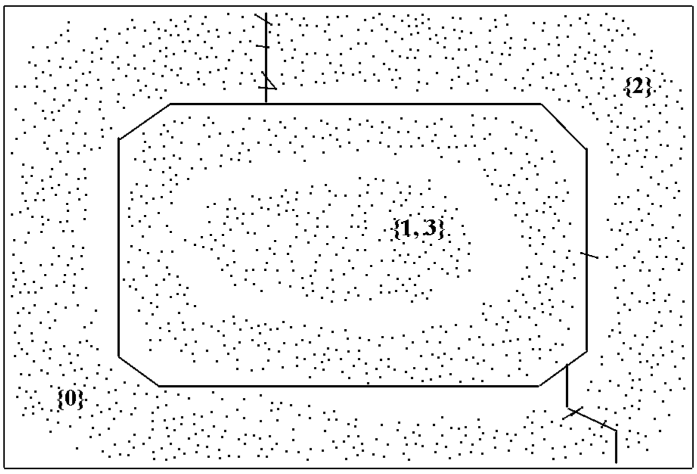

Fig. 6b. Decomposition L_3^3 found at run 2

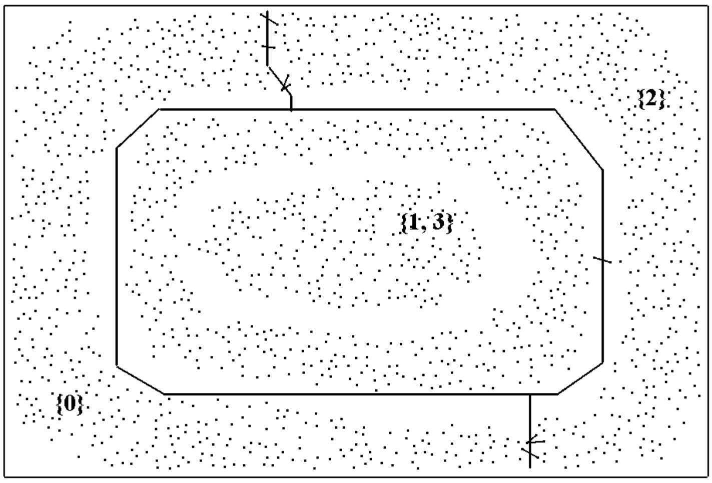

Fig. 6c. Decomposition L_3^3 found at run 3

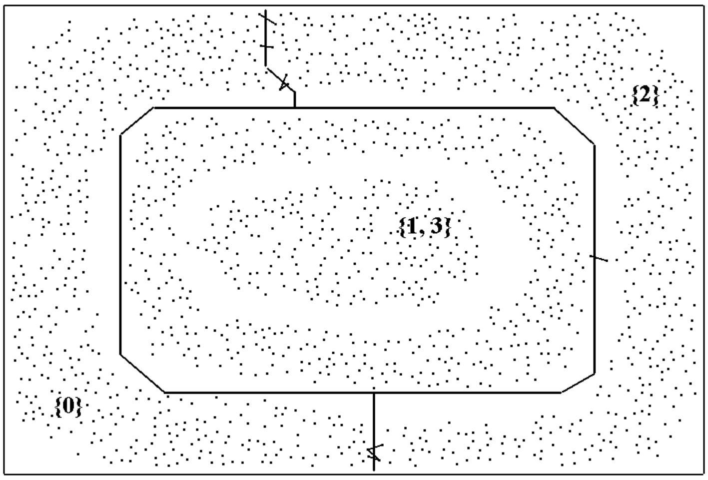

Fig. 6d. Decomposition L_3^3 found at run 4

It is easy to understand that in the same run essential decompositions L_4^4 are differing of the decompositions shown in Fig. 6 only in presence of another class in the center (compare also Fig. 4b and 4c). This implies that all these four decomposition also are different ones. At the same time essential decomposition L_2^2 and adjoined decomposition L_3^4 found at all the runs coincide with decompositions shown in Fig. 4a and 4d, i.e. they remain unchanged.

Thus, in the considered case the found family of decompositions $F(3, 4, 1)$ consists of 10 different decompositions. Among them, there are 8 varying with every run, and 2 permanent decompositions.

Example 8. Consider even more complicated case – the set of points, presented in Fig. 7a. The essential decompositions L_2^2, L_3^3, L_4^4, L_5^5, L_6^6, L_7^7, L_8^8, L_9^9 are shown in Fig. 7b – 7h. The numbers in these figures are the temporary names of the consecutively separated subgraphs. The greater peninsula is singled out after the 1st dichotomy, while the lesser is singled out only after the 7st dichotomy. The binary tree of the 11 consecutive dichotomies is shown in Fig. 8.

Computational experiments demonstrate that in different runs the greater peninsula is singled out always after the 1st dichotomy. At the same time the lesser peninsula is singled out at different steps of DAA, but always between 6th and 11th steps. Therefore, if we assume $k = 12$, then at every DAA run the lesser peninsula is singled out compulsory at some m^{th} step. Taking into account the agglomerative stage just after the construction of essential decomposition L_m^m (see subsection 2.2), we obtain the adjoined decomposition L_3^m into three subgraphs: two peninsulas and the remaining ring. This decomposition is shown in Fig. 9. The previous reasoning means that such decomposition is encountered at every DAA run. By the construction it means that the decomposition in Fig. 9 is the solution of the initial classification problem for the set, shown in Fig. 7a.

We can add that in three AC problems, considered in this Section 4, all the parameters of the algorithm were the same (including parameters of the graph dichotomy algorithm), except the number k of consecutive dichotomies. Pay attention that the notion of decomposition complexity is not used in the reasoning of the current Section 4. However, it seems intuitively that the AC problem of Example 6 is simpler than the AC problem of Example 7, and the AC problem of Example 7 is simpler than the AC problem of Example 8. It is true. Let us introduce the necessary notions and definitions.

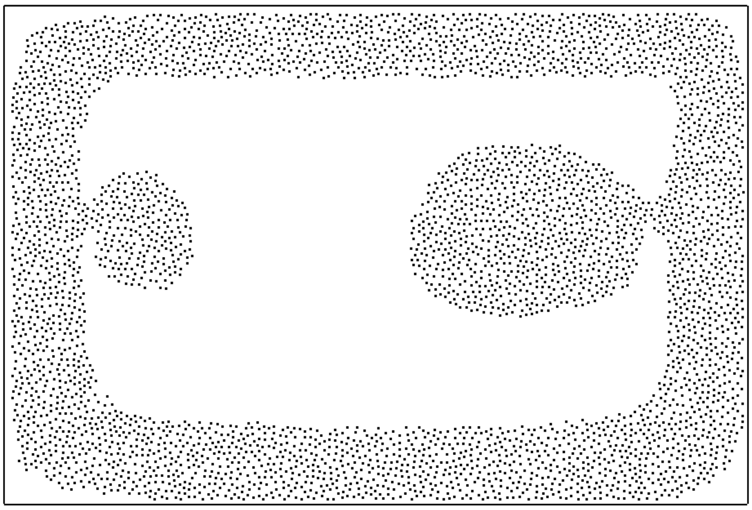

Fig. 7a. The initial set

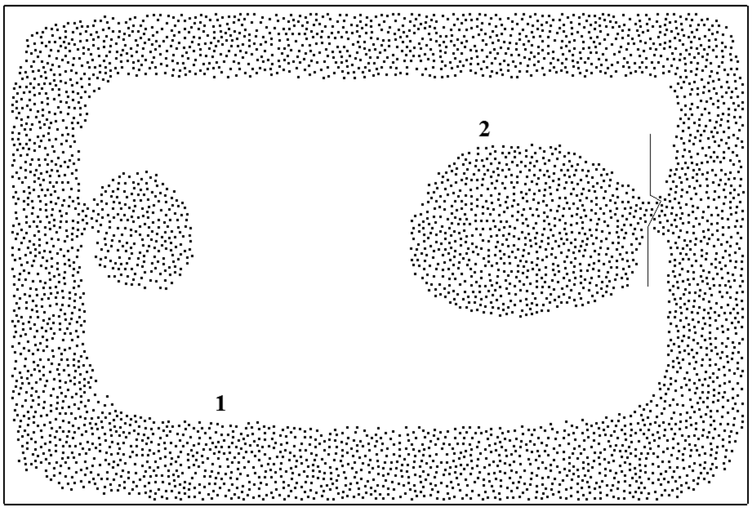

Fig. 7b. Essential decomposition L_2^2

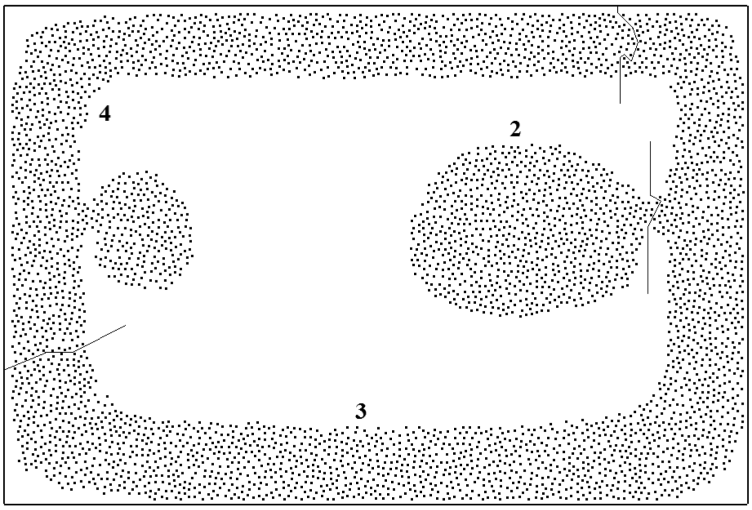

Fig. 7c. Essential decomposition L_3^3

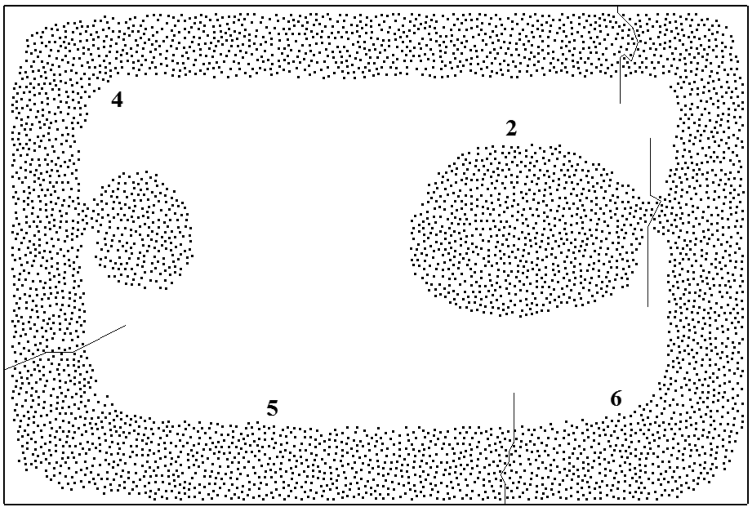

Fig. 7d. Essential decomposition L_4^4

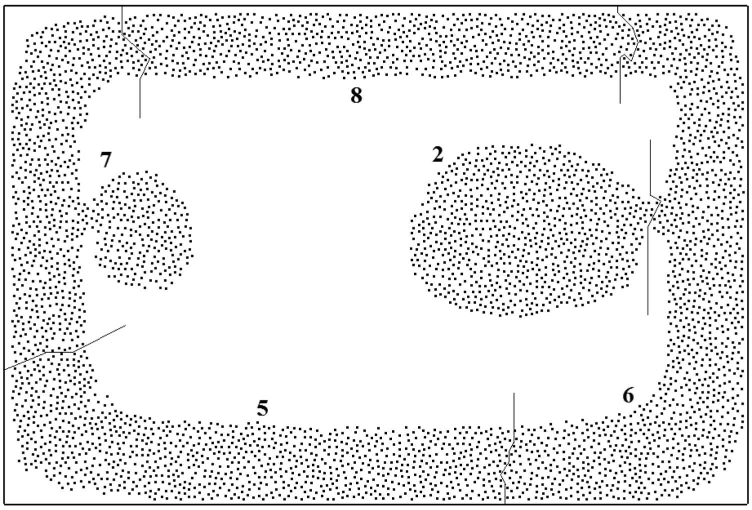

Fig. 7e. Essential decomposition L_5^5

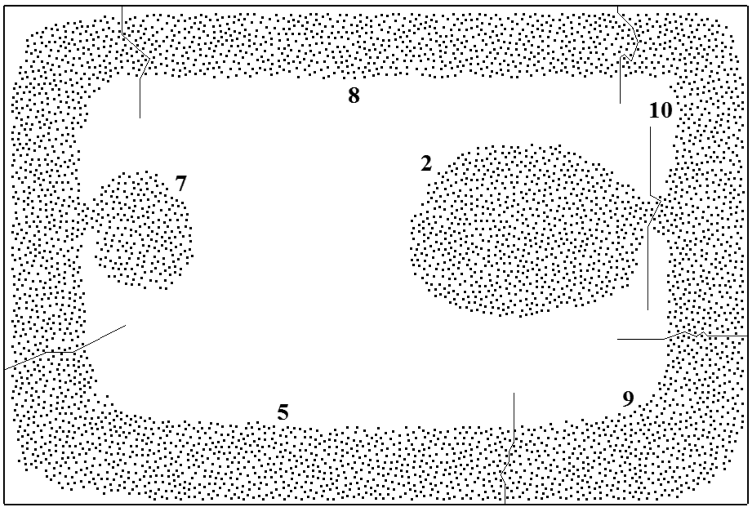

Fig. 7f. Essential decomposition L_6^6

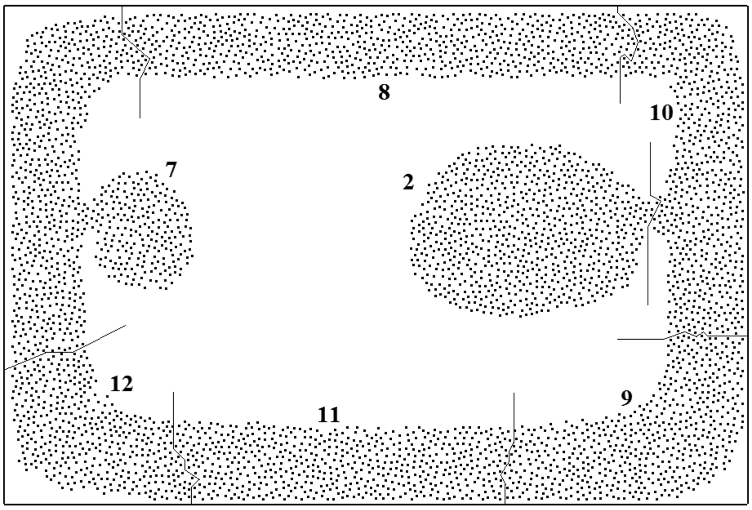

Fig. 7g. Essential decomposition L_7^7

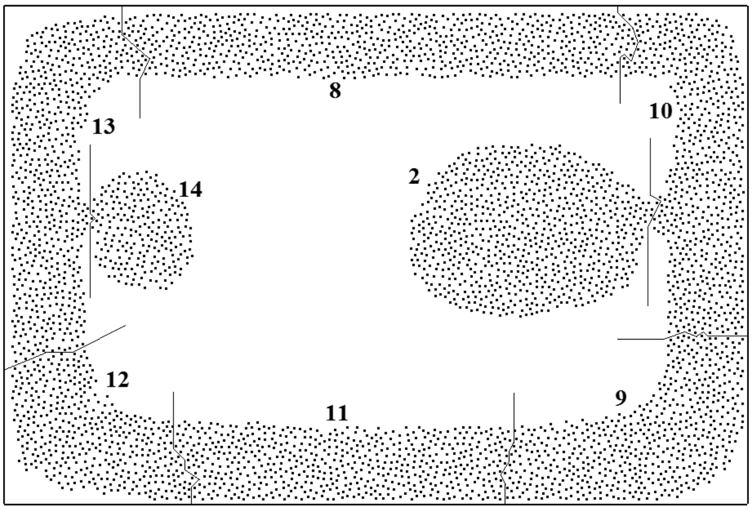

Fig. 7h. Essential decomposition L_8^8

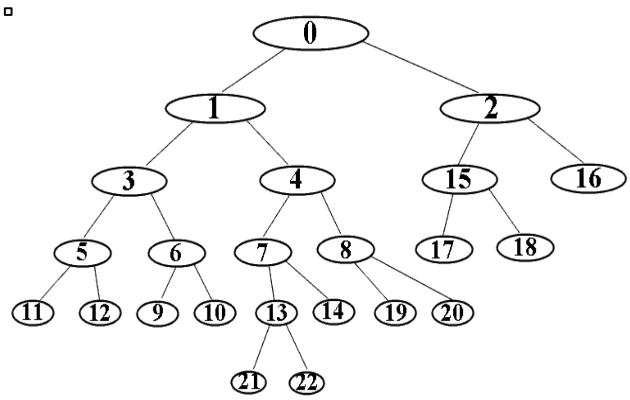

Fig. 8. Diagram of essential decompositions

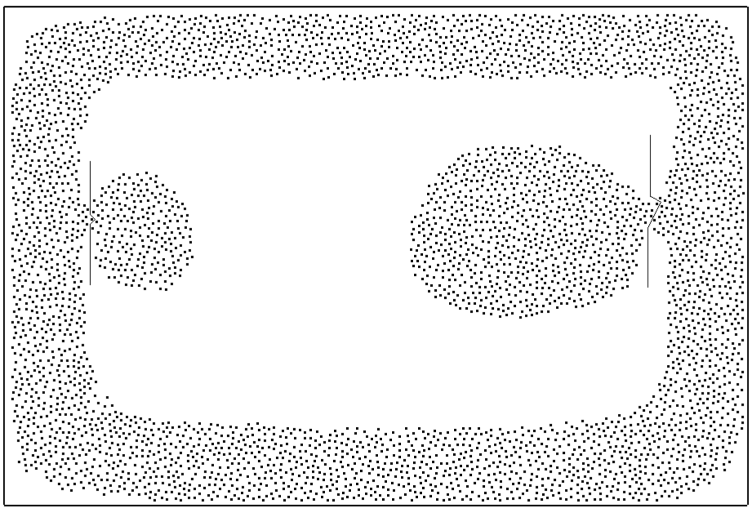

Fig. 9. Solution of initial classification problem in Fig. 7a

Remember that in the considered cases we need to do some number of consecutive dichotomies in DAA in order to find the correct classification. Denote by k^* the minimum value of parameter k, which guarantees construction of such a classification. This number can be considered as a ***complexity*** or ***difficulty*** of the initial AC problem. The corresponding values for the considered problems are presented in Table 1.

Complexity of Classification Problems

Table 1

	Example 6	Example 7	Example 8
k*	2	3	12

The numbers in Table 1 do not contradict to intuition. Of course, the definition in cases, there different numbers of consecutive dichotomies are required in different DAA runs, must be more accurate. It seems that it is possible to define k* as an average value over different DAA runs.

5. Analysis of Voting in RF Duma (Parliament)

The suggested approach to calculation of decomposition complexity of graph is applied here to analysis of voting in 2nd, 3rd and 4th RF Duma (1996 – 2007). First, the distance between every two deputies is defined, based on their votes. Second, analogously the construction in subsection 4.1, the distances between every two deputies determine a distance matrix and the corresponding neighborhood graph. Finally, the decomposition complexity index 1 for the constructed graph is related to a considered period of Duma activity. In more detail:

For every separate month of the considered period all the votes are considered. To every i^{th} deputy (i = 1, 2, …, m) a vector $v_i = (v_1^i, v_2^i, \ldots, v_n^i)$ is related, where n is the number of votes in a given month. Assume

$$v_j^i = \begin{cases} 1, \text{ if } i^{\text{th}} \text{ deputy voted for } j\text{th proposition;} \\ -1, \text{ if } i^{\text{th}} \text{ deputy voted against } j\text{th proposition;} \\ 0, \text{ otherwise (abstained or not participated).} \end{cases}$$

Note, that the number m of deputies, though slightly, changed from period to period. Of course, at every moment the number of deputies is always equal to 450. Yet during 4 years some deputies dropped out while the other ones came instead. The number of deputies participated in Duma voting activity in 1996-1997 was equal to 465, in 1998-1999 – to 485, in 2000-2003 – to 479 and in 2004-2007 – to 477.

Dissimilarity d_{st} between s^{th} and t^{th} deputies is defined as usual Euclidean distance between vectors v_s and v_t. The dissimilarity matrix $D = (d_{st})$ allows constructing an undirected graph G. For this graph G the corresponding family of decompositions were constructed and index 1 was calculated – just by the method, comprehensively described in Sections 2 and 3.

The following Table 2 presents the index 1 of complexity for every month of the voting activity of 2nd, 3rd and 4th RF Duma. The numbers in the 1st column are the dates (year and month). The numbers in the 2th column are equal to the number of votes in the corresponding months. Numbers in the 3rd column are equal to decomposition complexity 1 of the graph, calculated following the definition of this notion in Section 3. Here the number k of consecutive dichotomies is equal to 10, the number r of DAA runs also is equal to 10, parameter s = 1, so that the maximum number $\frac{(k+1)*k}{2} * r * s$ of decompositions is equal to 550. Empty rows correspond to months without any voting activity.

The numbers in the 3rd column in Table 2a – 2c, i.e. complexity of the graph, determined by voting results, demonstrate noticeable variability, though some trend are seen at once, by "unaided eye". Smoothed data, i.e. average value for half years, thereafter for years, and, finally, for whole period of every Duma activity, are presented in Table 3.

Complexity of Voting in 2nd Duma (1996–1999)

Table 2a

1	2	3	1	2	3
9601	174	0.610909	9801	248	0.421818
9602	321	0.625455	9802	366	0.330909
9603	295	0.581818	9803	347	0.469091
9604	470	0.683636	9804	334	0.436364
9605	263	0.938182	9805	292	0.398182
9606	269	0.827273	9806	489	0.534545
9607	450	0.263636	9807	493	0.352727
9608			9808		
9609			9809	405	0.390909
9610	432	0.494545	9810	326	0.507273
9611	226	0.567273	9811	338	0.327273

9612	566	0.465455		9812	534	0.392727
9701	234	0.456364		9901	416	0.207273
9702	427	0.445455		9902	354	0.250909
9703	334	0.381818		9903	482	0.369091
9704	437	0.316364		9904	384	0.372727
9705	169	0.485455		9905	228	0.449091
9706	762	0.238182		9906	768	0.392727
9707				9907		
9708				9908		
9709	337	0.201818		9909	292	0.241818
9710	354	0.247273		9910	338	0.270909
9711	253	0.289091		9911	696	0.218182
9712	530	0.265455		9912	243	0.430909

Complexity of Voting in 3rd Duma (2000–2003)

Table 2b

1	2	3		1	2	3
0001	71	0.547273		0201	279	0.183636
0002	228	0.112727		0202	380	0.063636
0003	177	0.387273		0203	311	0.081818
0004	368	0.112727		0204	640	0.114545
0005	279	0.141818		0205	353	0.138182
0006	454	0.149091		0206	956	0.072727
0007	301	0.078182		0207		
0008				0208		
0009	144	0.154545		0209	329	0.120000
0010	371	0.169091		0210	541	0.067273
0011	240	0.103636		0211	448	0.065454
0012	483	0.138182		0212	531	0.058182

0101	141	0.109091		0301	144	0.203636
0102	254	0.245455		0302	350	0.136364
0103	268	0.085454		0303	382	0.160000
0104	409	0.187273		0304	519	0.136364
0105	248	0.296364		0305	248	0.141818
0106	683	0.069091		0306	677	0.083636
0107	825	0.132727		0307		
0108				0308		
0109	200	0.140000		0309	208	0.221818
0110	360	0.069091		0310	428	0.072727
0111	668	0.160000		0311	400	0.203636
0112	600	0.101818		0312		

Complexity of Voting in 4th Duma (2004-07)

Table 2c

1	2	3		1	2	3
0401	101	0.360000		0601	168	0.216364
0402	220	0.101818		0602	204	0.289091
0403	270	0.141818		0603	256	0.265455
0404	295	0.101818		0604	255	0.147273
0405	249	0.325455		0605	179	0.194545
0406	385	0.143636		0606	365	0.085454
0407	378	0.372727		0607	260	0.221818
0408	268	0.303636		0608		
0409	101	0.274545		0609	230	0.114545
0410	252	0.261818		0610	305	0.278182
0411	355	0.349091		0611	528	0.320000
0412	535	0.250909		0612	463	0.260000
0501	130	0.283636		0701	243	0.214545
0502	209	0.421818		0702	189	0.356364

0503	237	0.225455
0504	355	0.090909
0505	255	0.123636
0506	300	0.338182
0507	240	0.141818
0508		
0509	174	0.325455
0510	266	0.360000
0511	359	0.232727
0512	426	0.225455

0703	262	0.123636
0704	368	0.187273
0705	190	0.118182
0706	448	0.169091
0707	320	0.310909
0708		
0709	141	0.167273
0710	350	0.298182
0711	337	0.227273
0712		

It seems that low value of complexity in 2002 was due to creation of the party "United Russia" and connected with attempts of straightening out the activity of Duma. It is surprising – at first sight – that in the 4th Duma in the condition of constitutional majority of this party the level of complexity is noticeably higher than in the 3rd Duma (0,235 opposite to 0,147), in which no party had majority. It is possible to say that for voting political bodies high complexity of corresponding graphs means inconsistence, maladjustment, irrationality of the whole body rather than individual fractions and deputies.

Smoothed complexity data

	Half 1	Half 2	Half 3	Half 4	Half 5	Half 6	Half 7	Half 8
Duma 2	0.711	0.448	0.387	0.251	0.432	0.394	0.340	0.290
Duma 3	0.242	0.129	0.165	0.121	0.109	0.078	0.144	0.166
Duma 4	0.196	0.302	0.247	0.257	0.199	0.239	0.195	0.251

	1st year	2nd year	3rd year	4th year
Duma 2	0.606	0.332	0.415	0.320
Duma 3	0.190	0.145	0.096	0.151
Duma 4	0.249	0.252	0.217	0.217

Duma 2	Duma 3	Duma 4
0.418	0.147	0.235

It is of interest to compare the data presented in Table 3b with the averaged for every year stability index for the 3rd Duma [Aleskerov et al, 2007]. These data, calculated using materials from the above-cited book, are presented in Table 4.

Stability index in the 3-rd Duma

Table 4

Year	2000	2001	2002	2003
Average stability index for one year	0,5597	0,5627	0,5339	0,5090

Maximum possible value of stability index is equal to 1, minimum possible value is equal to 0. In contrast to the complexity data, which has a clear-cut minimum in 2002, stability index does not reach the maximum in this year.

Note that most results in the book [Aleskerov et al, 2007] were obtained, basing on votes concerning only ***politically important issues***, while in the present work ***all the votes*** are used. Every approach has its advantages and disadvantages, whose detailed analysis is far beyond the framework of the present article. However, it is possible to suppose that the votes on politically important issues are much more controlled, than the votes on lesser important ones. It is one of cause of noticeable difference in some results. Of course, difference in the used formal methods of analysis also contributes in the above-mentioned difference in some conclusions. Particularly, the suggested approach gives highest level of complexity indices for voting bodies in two opposite cases – when all deputies vote unanimously and all the deputies vote at random. If the deputies are divided into two or more groups, which always vote correspondingly to their fixed political opinions, then the level of complexity indices is very low. It cannot be surprising after Examples 2 and 3.

Example 9. Let us consider families of decompositions of the graph constructed on voting of Duma deputies in May, 2001. The number r of runs is equal to 10. There are 10 essential decompositions L_4^4 (remember that every decomposition from L_4^4 consists by construction of 4 classes). Among these 10 decompositions there are only 3 different ones (the number before classes are equal to their cardinalities). These 3 decompositions are as follows.

Three different classifications
Classification 1

Class 1: 253 elements

1 3 4 5 21 22 23 24 25 26 27 28 30 34 39 40 43 44 45 47 48
53 55 56 57 58 61 62 64 66 67 69 70 74 76 77 78 79 81 82 83
84 85 89 90 91 92 93 94 96 97 100 102 103 108 109 110 111 113 114
115 119 121 122 124 126 128 129 134 137 139 143 147 148 155 157 159
162 165 167 168 170 174 175 179 180 181 182 183 186 188 193 195 197
199 200 202 203 205 208 209 210 214 216 217 218 220 221 222 225 227
229 230 231 232 236 238 239 240 245 246 248 251 252 253 254 255 256
257 258 259 262 266 268 271 272 273 274 275 278 281 282 285 286 287
288 292 296 298 299 300 301 303 305 307 311 316 317 318 319 320 321
322 324 326 331 335 336 337 338 339 340 343 344 346 347 352 355 358
361 363 364 365 366 367 369 370 373 375 376 377 378 379 380 381 386
388 393 394 397 398 400 404 405 407 410 411 414 415 417 420 421 422
424 425 429 431 432 433 437 440 441 444 445 446 448 449 450 451 454
455 456 457 458 460 461 462 463 464 465 466 467 468 469 470 471 472
473 474 475 476 477 478

Class 2: 79 elements

2 6 8 10 13 29 31 32 35 37 41 42 49 54 60 65 68 75 88 99 101
104 117 120 125 133 142 153 154 156 163 164 169 171 172 173 178 185
187 198 212 219 226 233 250 267 269 283 289 297 304 310 314 315 329
330 345 349 353 354 356 357 359 360 362 368 374 384 387 390 391 399
412 419 435 436 439 442 459

Class 3: 125 elements

0 7 9 11 12 15 16 18 19 20 33 36 38 46 50 51 52 59 63 71 72
73 80 86 87 95 98 105 106 107 112 118 127 130 131 132 135 138 140
141 144 146 149 150 151 152 158 160 161 166 176 177 184 189 190 191
192 194 196 201 204 207 211 215 224 228 234 235 237 242 243 260 261
263 264 265 270 277 279 280 284 290 291 293 295 302 306 308 309 312
313 323 325 327 328 332 333 334 341 342 348 350 351 371 382 383 385
389 392 395 396 401 402 403 406 408 409 416 423 426 427 430 443 447
453

Class 4: 22 elements

14 17 116 123 136 145 206 213 223 241 244 247 249 276 294 372 413 418
428 434 438 452

Classification 2

Class 1: 221 elements

1 3 4 5 21 22 23 24 26 27 28 30 34 40 44 45 47 48 53 55 56 57 58 61 64 67 69 70 74 77 78 79 82 84 85 89 90 91 92 93 94 96 97 100 102 103 108 109 110 111 113 114 115 119 121 122 124 126 128 129 134 137 139 143 147 148 155 157 159 162 165 167 168 174 175 179 180 181 182 183 186 188 195 197 199 200 202 203 208 209 210 214 216 217 218 220 221 222 225 227 229 230 231 232 236 238 239 240 246 248 251 252 253 257 259 262 266 268 271 272 273 274 275 278 281 282 285 286 287 288 292 298 299 300 301 303 307 311 316 319 320 321 324 326 331 335 337 338 339 340 343 344 346 347 352 355 358 361 364 365 366 367 370 373 376 377 380 381 386 388 393 394 397 398 400 404 405 407 410 411 415 417 420 421 422 424 431 432 437 440 441 444 445 446 448 449 450 454 455 456 457 458 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478

Class 2: 79 elements

2 6 8 10 13 29 31 32 35 37 41 42 49 54 60 65 68 75 88 99 101 104 117 120 125 133 142 153 154 156 163 164 169 171 172 173 178 185 187 198 212 219 226 233 250 267 269 283 289 297 304 310 314 315 329 330 345 349 353 354 356 357 359 360 362 368 374 384 387 390 391 399 412 419 435 436 439 442 459

Class 3: 125 elements

0 7 9 11 12 15 16 18 19 20 33 36 38 46 50 51 52 59 63 71 72 73 80 86 87 95 98 105 106 107 112 118 127 130 131 132 135 138 140 141 144 146 149 150 151 152 158 160 161 166 176 177 184 189 190 191 192 194 196 201 204 207 211 215 224 228 234 235 237 242 243 260 261 263 264 265 270 277 279 280 284 290 291 293 295 302 306 308 309 312 313 323 325 327 328 332 333 334 341 342 348 350 351 371 382 383 385 389 392 395 396 401 402 403 406 408 409 416 423 426 427 430 443 447 453

Class 4: 54 elements

14 17 25 39 43 62 66 76 81 83 116 123 136 145 170 193 205 206 213 223 241 244 245 247 249 254 255 256 258 276 294 296 305 317 318 322 336 363 369 372 375 378 379 413 414 418 425 428 429 433 434 438 451 452

Classification 3

Class 1: 243 elements

1 3 4 5 14 17 21 22 23 24 26 27 28 30 34 40 44 45 47 48 53 55 56 57 58 61 64 67 69 70 74 77 78 79 82 84 85 89 90 91 92 93 94 96 97 100 102 103 108 109 110 111 113 114 115 116 119 121 122 123 124 126 128 129 134 136 137 139 143 145 147 148 155 157 159 162 165 167 168 174 175 179 180 181 182 183 186 188 195 197 199 200 202 203 206 208 209 210 213 214 216 217 218 220 221 222 223 225 227 229 230 231 232 236 238 239 240 241 244 246 247 248 249 251 252 253 257 259 262 266 268 271 272 273 274 275 276 278 281 282 285 286 287 288 292 294 298 299 300 301 303 307 311 316 319 320 321 324 326 331 335 337 338 339 340 343 344 346 347 352 355 358 361 364 365 366 367 370 372 373 376 377 380 381 386 388 393 394 397 398 400 404 405 407 410 411 413 415 417 418 420 421 422 424 428 431 432 434 437 438 440 441 444 445 446 448 449 450 452 454 455 456 457 458 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478

Class 2: 79 elements

2 6 8 10 13 29 31 32 35 37 41 42 49 54 60 65 68 75 88 99 101 104 117 120 125 133 142 153 154 156 163 164 169 171 172 173 178 185 187 198 212 219 226 233 250 267 269 283 289 297 304 310 314 315 329 330 345 349 353 354 356 357 359 360 362 368 374 384 387 390 391 399 412 419 435 436 439 442 459

Class 3: 125 elements

0 7 9 11 12 15 16 18 19 20 33 36 38 46 50 51 52 59 63 71 72 73 80 86 87 95 98 105 106 107 112 118 127 130 131 132 135 138 140 141 144 146 149 150 151 152 158 160 161 166 176 177 184 189 190 191 192 194 196 201 204 207 211 215 224 228 234 235 237 242 243 260 261 263 264 265 270 277 279 280 284 290 291 293 295 302 306 308 309 312 313 323 325 327 328 332 333 334 341 342 348 350 351 371 382 383 385 389 392 395 396 401 402 403 406 408 409 416 423 426 427 430 443 447 453

Class 4: 32 elements

25 39 43 62 66 76 81 83 170 193 205 245 254 255 256 258 296 305 317 318 322 336 363 369 375 378 379 414 425 429 433 451

Classifications 1 и 2 are encountered 4 times from 10, classification 3 – 2 times. The other classifications do not arise for any number r of repetitions.

6. Stock Market Analysis

The stock market S&P-500 (500 greatest companies in USA) is considered. First of all let us describe the graph model of this stock market (for other stock markets it can be done analogously; see [Boginski, Butenko, Pardalos, 2005]). The objects correspond to considered (during some period) shares. The distance between two shares is determined as follows.

1. Let us define the basic minimal period, consisting of l consecutive days. All the data found for the period x, $x-1$, … , $x-l+1$ are related to day x. Assume the length l of a considered period is equal to 16. This choice is determined by the following meaningful reasons: for short period data are too variable, for long period – too smooth. The choice of parameters in general is discussed in the Conclusion.

2. Prices of all the shares at closure time are considered for days x, $x-1$, … , $x-l+1$. The matrix R of pairwise correlation coefficients is calculated basing on these prices.

3. Distance d_{ij} between two shares (say, i and j) is defined by the formula $d_{ij} = 1-r_{ij}$, where r_{ij} is the correspondent element of matrix R. The determined distance d is close to 0 for «very similar» shares and is close to 2 for «very dissimilar» shares. Therefore matrix $D = (d_{ij})$ is considered as the dissimilarity matrix. Following the material of the subsection 4.1, reduce the dissimilarity matrix to the undirected graph G, whose vertices correspond to the shares in the stock market S&P-500. Pay attention that these sets of shares can be different for different last days x of a considered period.

As before in Sections 4 and 5, we can find family of the graph decompositions by the way, described in Section 2, and after that calculate the complexity index 2 as it is described in Section 3. In order to do it we must determine all the necessary parameters. Let the number k of successive dichotomies is equal to 2 (see subsection 2.2). It means that the initial set of vertices is divided into 2 subgraphs, thereafter the larger of these subgraphs also is divided into 2 subgraphs and finally 2 subgraphs of the 3 ones, connected by the maximum number of edges, are pooled to one subgraph (see Example 6). Thus, three classifications – one into 3 classes and two into 2 classes – are constructed. The two latest can coincide or can be different.

Number r of runs in inner loop is equal to 150, number s of repetitions of the corresponding inner loop cycle (see subsection 2.3) is equal to 10. It means that family $F(2,150,10)$ consists of $4500 = 3\times150\times10$ classifications, some of which can coincide. It is possible to calculate the complexity index 2 of family $F(2,150,10)$ (see formulae (6) and (7)). This number (entropy of family $F(2,150,10)$) is related to day x (the last day of a 16-days period. Therefore, $E(x)$ can be determined in the evening of day x – practically in several minutes after closure time.

The results of two independent calculation of $E(x)$ for one day (specifically, 01.01.2001) are presented in Table 5. The found values are very close (they differ approximately in 0.002), despite the significantly larger differences of values over 150 runs, presented in both rows of the table. This demonstrates stability of the application of two-dimensional averaging scheme, discussed in Section 3.

Two-Dimensional Averaging Scheme

Table 5

Number of outer loop i	1	2	3	4	5	6	7	8	9	10	Average value
Entropy $E(3,i,150)$	5.335	5.396	5.328	5.323	5.367	5.345	5.358	5.323	5.353	5.392	5.352
Entropy $E(3,i,150)$	5.300	5.385	5.357	5.321	5.383	5.356	5.375	5.354	5.360	5.306	5.350

Let us consider the period since 01.01.2001 until 31.12.2010. This period includes two big crises: dotcom crisis in 2001 and hypothec crisis (become world crisis) in 2008. The entropy $E(x)$ is calculated for every day x from the considered 3652-days period. The ave-rage values of entropy at every day are stable enough, as well as in the 1st day of the period (see table 3).

Entropy $E(x)$ at every day x is presented in Table 6. For commodity, results for every year are given separately. Some groups of seven consecutive days are marked by gray background. It will be explained in the next subsection 6.1.

Entropy at every day in 2001 – 2010

Table 6

2001

5.353	5.229	6.023	5.020	4.904	5.115	4.991	5.348	5.682	4.638	4.622	4.968
4.879	4.680	4.623	5.977	5.446	6.077	5.227	6.061	6.041	5.949	5.371	5.386
5.929	5.894	5.317	5.611	5.656	5.047	6.091	5.531	4.711	4.613	5.223	5.438
5.330	5.483	5.216	5.366	5.382	5.658	4.922	5.203	5.524	5.556	5.725	5.428
5.477	5.253	5.786	5.045	5.580	5.566	5.694	4.979	4.887	5.854	5.981	5.757
6.028	5.351	5.175	5.362	5.877	4.662	4.835	5.750	5.387	5.687	5.442	5.500
5.780	5.481	5.356	5.431	4.633	4.457	4.506	4.742	4.935	4.726	5.119	4.543
4.448	5.537	5.640	5.265	6.091	6.011	4.552	4.765	4.626	5.439	5.064	5.102
5.527	5.892	5.999	5.863	6.091	5.935	5.809	5.875	4.745	5.211	5.655	5.186
4.702	5.949	5.772	5.668	5.289	4.950	5.421	5.290	4.602	4.757	5.000	5.053
6.074	5.828	5.452	5.984	5.424	5.219	5.196	4.992	5.218	5.495	5.865	5.933
5.795	5.600	5.443	5.593	5.524	5.780	5.528	5.841	5.014	4.978	4.736	4.813
4.807	4.966	5.395	5.289	4.975	5.755	5.903	5.964	5.146	5.137	5.593	5.612
5.408	5.594	5.122	5.383	5.559	5.098	5.829	5.938	5.713	5.569	5.786	5.369
5.214	4.650	4.700	6.028	4.868	4.943	5.520	5.391	4.805	5.846	5.705	6.057
5.733	5.424	4.800	5.189	5.974	5.798	5.153	5.505	4.917	5.492	5.481	4.830
5.604	5.855	5.924	5.216	5.702	5.613	5.597	5.278	5.838	5.222	4.593	5.062
5.551	5.642	5.834	4.735	5.546	5.777	6.060	5.870	5.667	5.096	5.002	5.137
5.956	5.876	5.926	4.885	6.075	5.724	6.069	5.872	5.133	5.818	5.486	4.663
5.204	5.134	5.106	4.845	5.512	5.565	5.079	5.479	4.861	5.775	6.003	5.901
5.523	5.942	5.393	5.226	4.667	4.863	4.688	5.216	4.667	5.601	4.994	4.623
4.439	4.530	4.960	4.541	5.180	5.172	4.710	5.312	5.860	4.492	4.500	4.497
4.468	4.460	4.457	4.528	4.472	5.677	4.920	5.502	5.972	5.482	5.377	5.737
5.729	5.674	5.973	6.077	4.677	5.263	5.594	5.130	4.936	4.823	4.978	4.513
5.485	5.619	5.172	5.422	5.839	6.000	5.838	5.971	5.230	5.204	6.087	5.956
5.641	5.132	5.323	5.213	5.339	4.894	5.525	5.408	4.539	5.551	5.003	5.288
4.891	5.046	4.710	5.126	5.455	5.542	4.496	5.172	5.672	5.141	4.650	4.610
5.889	4.831	5.228	4.822	5.162	4.630	5.326	5.619	6.090	5.926	5.985	6.001
5.640	6.080	5.452	5.457	5.643	5.629	4.918	4.874	4.836	5.407	4.726	5.674
5.649	6.046	4.883	5.745	5.846	6.038	5.915	5.066	5.625	5.659	4.936	5.058
5.013	5.213	5.069	4.986	4.760							

2002

5.279	6.029	5.993	5.652	5.739	5.745	5.696	5.604	5.208	5.217	5.842	4.981
5.676	4.975	5.437	5.218	4.882	5.126	5.694	4.942	5.030	4.902	5.167	4.732
5.120	4.869	5.944	6.067	5.064	4.705	5.469	5.104	5.569	5.357	5.450	5.161
5.834	5.760	5.128	5.037	5.127	5.984	5.492	6.025	5.709	6.052	5.915	5.702
5.793	5.192	5.696	5.253	5.831	5.138	5.705	5.606	5.589	5.605	5.110	5.876

5.310	5.584	5.446	4.916	4.549	5.049	4.567	4.577	4.518	4.581	5.603	4.984
5.939	5.232	4.951	4.825	6.043	5.113	5.175	5.432	5.532	5.018	5.996	5.476
5.517	5.841	5.711	5.765	5.838	4.736	5.906	5.728	5.209	5.923	6.073	5.923
5.641	5.025	4.832	5.564	5.518	5.082	4.594	4.616	4.967	5.681	5.057	5.890
5.417	5.623	5.592	5.728	5.993	5.864	5.418	5.135	5.165	4.758	4.614	5.050
4.763	5.296	5.929	5.273	5.109	5.100	4.905	6.017	5.756	5.977	6.029	5.913
5.237	5.636	5.599	6.078	5.333	5.718	5.795	5.600	5.672	5.594	4.698	5.411
5.909	5.973	5.844	5.773	4.712	5.126	5.232	5.290	6.093	5.641	4.599	5.178
4.650	4.587	4.554	4.559	4.734	4.917	4.712	5.481	5.143	5.307	5.330	5.388
6.057	5.815	4.857	5.207	5.021	5.099	5.768	5.689	4.930	5.504	5.101	4.979
5.359	5.238	5.101	4.562	4.973	4.707	5.080	4.835	4.744	5.337	4.596	5.950
4.829	6.078	5.910	5.017	5.452	4.984	4.439	4.499	4.694	4.736	4.465	4.450
4.524	4.486	5.063	5.390	5.513	4.735	5.218	4.644	4.877	4.608	4.684	4.791
5.388	5.113	5.638	5.270	5.108	4.959	5.057	5.474	5.452	4.443	5.494	4.439
5.036	4.499	4.443	4.762	5.257	4.855	5.598	5.210	4.821	4.676	5.000	6.053
5.024	5.918	4.920	5.189	4.827	5.261	5.525	4.837	4.746	5.833	6.079	4.846
5.697	5.525	5.912	5.464	5.632	5.787	5.758	5.952	5.964	5.807	5.964	5.984
5.364	4.507	5.025	4.480	4.541	5.136	5.624	5.707	4.654	5.112	5.334	6.008
5.941	4.569	4.776	4.866	5.306	5.845	5.160	5.030	5.155	5.373	6.047	5.814
5.572	5.423	5.055	4.575	4.605	4.517	4.899	5.060	4.902	4.985	5.835	5.327
4.790	5.658	6.081	5.962	6.013	6.019	6.052	5.860	5.484	5.279	5.808	5.913
5.057	4.982	5.476	5.315	5.310	6.067	4.883	6.070	6.092	6.084	5.992	6.063
4.925	5.410	4.679	5.037	5.595	4.786	4.745	4.511	4.852	5.259	5.724	5.104
5.012	5.653	6.073	6.093	5.932	5.950	4.787	4.838	4.996	5.322	4.744	5.090
4.981	5.683	5.152	5.261	5.753	5.734	5.628	5.571	5.874	4.439	5.475	5.675
6.085	5.500	5.845	5.382	5.472							

2003

5.771	5.762	5.489	5.763	4.733	5.613	4.633	5.169	4.744	4.801	4.758	4.842
4.816	5.629	5.186	6.031	5.945	5.524	5.706	5.863	5.413	4.604	5.118	5.297
4.904	4.566	5.295	4.520	4.564	4.628	4.891	5.564	4.946	5.137	5.834	5.564
4.810	5.165	5.748	5.445	5.231	5.858	4.802	4.542	5.126	5.262	4.728	4.744
5.005	5.661	5.978	5.205	6.019	4.668	6.067	5.578	5.725	4.985	5.461	5.682
5.576	5.467	5.829	4.911	5.971	5.835	5.841	5.808	5.349	4.859	4.981	5.345
4.550	4.532	4.639	6.077	5.752	5.728	4.866	4.772	5.579	4.667	4.439	4.520
4.503	5.907	5.453	5.324	5.416	5.511	5.711	4.596	5.169	5.348	4.698	5.712
5.898	5.616	5.258	5.567	5.764	4.911	5.348	5.668	5.316	4.452	4.817	5.035
5.387	5.760	5.360	4.903	5.626	4.882	5.270	4.996	4.895	5.788	5.989	5.111
5.945	5.400	5.377	5.393	5.446	4.994	5.174	4.871	5.749	5.686	5.177	4.439
5.191	5.302	5.494	4.826	5.511	5.416	5.136	5.029	4.830	5.256	6.053	5.885
5.924	5.903	5.725	5.554	5.811	5.065	5.538	4.597	4.642	4.798	4.452	4.949
5.984	5.243	5.238	5.622	4.777	4.493	4.699	4.722	5.079	5.770	5.716	5.840
5.683	5.853	5.824	5.765	5.353	5.637	5.598	5.976	5.485	5.237	5.395	5.902
5.711	5.058	5.169	5.206	5.150	5.508	5.285	5.319	4.952	5.587	5.324	4.993
4.532	4.593	4.635	4.739	4.935	5.699	4.498	5.406	5.789	4.842	5.817	5.658
5.239	5.496	5.236	6.034	6.053	5.927	6.039	6.027	5.199	5.286	5.983	5.973
5.487	4.994	5.743	6.106	6.046	5.360	5.773	6.039	5.692	5.478	5.651	5.519

5.689	5.846	5.938	5.993	5.267	4.762	4.854	4.617	5.204	5.150	5.412	5.970
5.790	5.423	5.661	4.746	6.087	5.346	4.643	4.678	5.365	5.173	5.434	5.611
5.014	5.267	5.372	5.791	5.380	5.613	4.977	5.032	5.515	5.858	6.058	5.511
6.029	5.514	5.170	5.502	4.589	5.567	5.340	5.288	5.957	5.508	5.761	5.449
4.705	4.661	5.782	5.578	4.693	5.232	4.856	4.612	4.593	5.121	4.492	4.508
4.706	4.821	5.227	5.732	5.136	5.440	5.595	5.276	5.896	6.053	6.016	5.252
5.263	5.323	6.066	5.351	5.592	6.013	4.990	4.722	5.671	4.716	5.402	4.930
5.978	4.910	5.470	5.540	5.193	5.756	5.289	5.003	5.089	5.632	5.496	5.181
4.439	5.202	5.681	5.010	4.819	5.549	5.077	5.763	5.180	4.463	4.511	5.508
5.560	4.528	5.465	5.111	4.963	5.054	5.443	5.318	4.680	5.492	5.337	4.820
4.862	5.030	5.731	5.814	5.861	6.063	4.954	5.664	5.915	5.619	5.347	4.687
5.326	5.602	5.829	4.612	5.156							

2004

4.799	5.942	5.072	4.886	5.578	6.076	5.349	5.209	5.015	5.145	5.433	5.210
5.985	5.768	5.244	5.960	5.457	5.308	5.580	5.304	5.297	5.314	5.788	4.694
4.831	4.685	5.240	5.348	5.501	5.954	5.282	5.071	5.213	5.801	5.160	5.237
5.439	5.085	4.919	5.864	5.477	5.844	5.397	6.036	4.556	5.592	5.635	5.280
5.397	6.080	5.779	5.352	5.583	5.370	5.799	5.396	5.143	5.893	5.694	5.120
5.799	5.882	5.330	4.733	4.946	5.519	4.922	5.192	4.863	5.882	5.499	5.992
5.611	5.577	4.753	5.313	5.659	5.538	5.520	5.408	4.840	5.021	4.825	4.635
5.704	5.411	5.181	5.395	5.733	5.988	5.440	5.537	5.432	5.291	5.084	4.730
4.751	4.871	4.554	4.463	4.659	4.572	4.439	5.169	5.507	5.811	4.736	5.180
4.809	5.329	5.634	5.406	5.820	5.407	5.205	4.853	5.780	6.038	5.969	5.336
5.571	5.981	5.935	5.568	5.955	5.400	5.555	5.016	5.562	4.955	5.068	5.188
4.439	4.789	4.650	4.573	5.114	4.863	4.959	5.275	6.052	4.899	5.594	5.821
5.141	5.925	5.745	5.683	5.605	5.420	4.986	5.052	5.130	5.047	4.965	6.001
6.043	5.236	5.803	5.575	5.418	6.031	5.795	5.114	5.666	6.036	6.091	4.821
5.905	5.541	5.108	5.070	5.368	4.971	4.839	5.332	5.220	4.606	4.970	5.284
4.955	5.842	5.841	5.584	5.814	5.640	5.492	5.978	5.894	4.728	5.241	5.391
4.658	5.192	5.474	4.605	5.893	5.583	5.172	4.578	4.602	5.461	5.376	5.993
5.181	5.802	5.791	5.485	5.730	4.671	5.167	4.818	4.855	5.390	5.912	5.662
6.024	5.787	5.653	4.905	4.665	4.789	4.927	4.535	4.618	4.452	5.048	4.600
5.076	4.956	5.083	5.126	5.521	5.261	5.636	4.981	4.643	4.439	4.637	4.784
4.876	5.033	4.881	5.229	5.923	4.791	4.600	5.318	4.823	4.798	4.930	4.744
5.583	4.887	4.850	4.515	5.647	4.988	5.188	5.603	5.761	5.670	5.900	5.266
5.130	5.907	5.452	5.501	5.841	5.308	4.818	5.350	5.602	5.417	5.184	5.286
4.898	5.431	5.133	5.125	5.532	4.588	4.712	4.463	4.650	5.448	5.045	5.158
5.783	5.193	5.016	4.896	4.861	5.397	4.717	5.821	4.809	5.546	5.440	5.613
5.425	5.050	5.693	5.458	5.017	4.629	5.202	4.498	4.640	4.472	4.439	4.502
4.439	4.650	4.527	4.967	5.211	4.830	5.699	5.139	5.358	5.527	5.717	4.941
5.512	5.171	4.439	4.770	5.954	5.087	4.899	5.507	5.658	5.500	5.630	5.399
5.376	5.283	4.884	4.602	4.763	5.543	5.641	4.999	5.020	5.497	5.586	4.967
6.006	5.104	5.484	5.807	6.042	6.036	6.059	5.233	5.092	5.995	5.890	5.630
5.644	5.359	5.519	4.863	5.573	5.029						

2005

5.275	4.650	5.819	5.255	5.675	5.857	4.739	4.862	4.463	4.563	4.576	4.763
5.303	5.023	5.279	5.823	5.995	4.777	5.754	4.775	5.139	5.495	5.568	5.856
5.877	5.690	5.763	5.473	5.287	4.777	4.439	5.469	5.645	5.279	5.052	4.675
5.352	4.901	4.930	5.388	4.815	4.576	5.019	4.587	5.028	4.545	4.685	5.124
5.584	5.111	5.118	5.485	5.715	5.408	5.166	5.986	5.171	5.680	4.815	5.828
5.546	5.288	5.719	4.601	4.636	4.519	4.748	4.919	5.391	5.163	6.031	5.913
5.534	4.615	5.523	5.678	5.349	5.618	5.505	5.594	5.774	5.015	5.056	5.024
5.115	5.018	5.121	5.125	5.778	4.845	6.007	5.131	4.904	4.439	6.038	5.573
5.552	4.672	4.870	5.685	5.735	5.616	5.886	5.386	5.273	4.610	4.665	5.078
4.439	4.439	4.617	4.439	4.711	5.618	5.446	5.976	5.387	5.952	6.093	5.335
5.102	5.913	5.824	5.064	5.962	5.257	5.079	5.119	5.233	5.474	5.828	4.917
5.881	5.759	4.680	5.168	6.066	5.116	5.674	5.747	5.511	5.364	5.227	5.059
5.273	4.864	4.472	4.929	5.201	5.063	5.423	5.934	5.204	5.537	5.748	5.825
5.249	4.880	5.459	5.655	5.235	5.568	5.292	5.198	5.894	5.807	5.981	5.291
5.003	5.427	5.614	5.606	4.643	5.143	5.815	5.616	5.578	4.629	4.695	5.275
5.046	4.702	5.419	5.113	4.863	5.756	5.101	5.908	5.648	4.876	4.439	4.610
4.914	4.983	5.639	5.105	4.452	4.765	5.186	5.081	5.161	5.148	5.422	5.429
5.512	5.298	5.711	5.809	5.812	4.609	5.297	5.380	5.547	5.049	5.335	5.297
5.874	5.831	5.261	4.968	5.049	4.754	5.616	5.952	4.637	5.328	5.262	5.250
5.297	5.592	4.814	5.856	4.545	4.439	5.621	5.651	5.208	5.522	4.861	4.962
4.975	5.551	5.372	5.854	5.413	5.928	5.346	5.404	5.017	5.295	4.870	5.622
4.439	5.415	5.136	5.019	4.701	4.994	4.710	5.918	5.493	5.257	5.384	5.307
5.295	5.372	4.984	5.414	4.687	5.114	5.276	5.686	5.589	5.263	5.421	4.492
5.254	5.786	5.941	5.886	5.422	5.297	5.716	4.802	4.622	4.622	4.604	4.520
4.895	5.616	5.093	4.947	5.918	4.770	5.211	5.095	5.025	5.214	5.507	5.910
5.926	5.773	5.397	5.651	5.055	5.248	5.649	4.693	5.063	4.587	4.971	5.707
5.007	5.454	5.385	4.930	5.058	5.324	5.486	5.800	4.867	5.692	4.852	4.931
5.238	4.658	4.705	5.173	4.963	4.515	4.439	5.214	4.914	5.025	5.210	4.690
4.972	5.555	4.739	6.041	5.576	5.540	5.542	5.165	6.082	5.982	5.486	5.345
5.844	5.854	5.399	4.930	5.948	5.351	6.054	5.374	5.129	5.922	5.878	5.973
5.224	5.720	5.544	5.127	5.091							

2006

5.438	5.264	4.909	5.984	5.672	5.492	5.288	5.560	6.057	4.480	4.797	5.914
5.255	4.839	4.929	4.717	5.031	5.578	5.175	4.780	5.404	5.597	5.543	5.311
5.778	5.778	5.777	4.899	4.691	5.548	4.439	4.545	6.017	5.787	5.979	5.652
5.625	5.411	5.383	6.066	5.804	5.261	5.287	6.071	5.069	5.935	5.255	5.503
5.854	5.577	5.181	5.612	5.181	5.785	5.082	4.875	5.227	5.156	5.415	5.410
5.265	5.694	4.687	4.439	5.880	5.373	5.415	5.447	6.073	5.537	5.537	5.093
4.973	4.919	5.644	5.400	6.011	5.826	5.303	5.065	4.616	5.464	4.913	5.091
5.140	5.325	5.534	5.209	5.786	6.040	5.554	5.942	5.168	5.896	5.703	5.746
5.412	5.655	4.892	4.898	5.290	5.528	5.553	4.875	5.089	4.823	5.882	5.835
5.941	5.988	6.038	5.238	5.014	4.916	5.425	4.643	4.795	4.919	5.812	5.747
5.714	5.630	4.900	5.515	6.060	5.951	5.763	4.744	5.312	5.502	5.964	5.815
5.362	5.520	4.970	5.196	4.727	5.043	4.661	4.898	5.172	5.321	4.705	4.452

5.423	5.074	5.878	6.009	5.777	5.578	5.310	6.000	4.812	4.848	5.244	5.846
4.674	5.541	6.018	4.666	5.047	4.774	4.957	4.833	5.868	4.862	5.417	4.851
5.091	4.958	6.056	6.073	5.876	6.093	6.022	6.064	5.870	5.634	5.561	6.102
5.685	4.579	5.117	4.909	5.214	4.687	4.963	4.759	5.760	5.731	5.751	5.420
5.461	4.715	5.102	5.162	4.480	4.928	4.439	5.573	5.013	4.903	5.127	4.770
4.761	4.649	4.974	5.893	4.901	5.336	4.439	4.761	4.993	4.949	4.898	4.729
5.165	5.027	5.438	5.875	5.678	6.106	5.211	5.270	6.044	4.920	5.572	5.165
5.952	5.036	5.256	4.972	5.347	4.775	4.439	4.862	5.302	6.025	5.885	5.570
5.883	5.049	5.194	5.995	5.704	5.813	5.754	4.559	5.215	4.740	5.461	5.065
5.886	5.502	5.996	5.847	5.520	5.847	5.182	4.797	4.917	5.057	5.173	4.951
4.980	5.559	5.576	5.993	5.835	6.037	5.965	6.031	6.013	5.419	5.298	5.605
5.066	5.548	5.977	5.360	5.330	5.475	5.022	4.895	5.402	5.250	4.734	5.022
5.535	5.419	4.904	4.855	5.453	5.344	5.905	5.896	5.372	5.371	5.282	5.701
4.654	5.103	5.714	4.858	5.878	5.900	6.085	5.293	5.956	6.065	5.799	5.093
5.391	4.986	5.203	5.912	6.013	5.422	5.094	5.793	5.711	6.096	5.974	6.005
5.369	5.331	5.761	5.633	5.580	5.299	5.535	5.875	5.519	5.074	5.088	4.853
4.937	4.836	4.889	4.912	5.766	6.036	4.927	5.474	4.972	5.649	5.516	5.077
4.899	5.663	5.338	5.928	5.638	5.816	5.474	5.341	5.332	4.993	5.785	5.031
4.961	5.571	5.018	4.488	4.439							

2007

5.061	5.778	5.382	5.905	5.993	6.019	6.018	5.173	4.838	5.689	4.855	4.782
5.148	5.197	5.387	4.962	4.896	4.679	4.925	5.319	4.835	5.551	4.921	4.602
5.191	4.954	5.854	5.572	5.093	5.989	5.050	5.268	5.320	5.016	4.909	4.887
4.660	4.853	4.680	4.835	5.017	4.614	4.777	5.335	5.938	6.009	5.804	4.965
5.046	5.646	5.150	5.038	5.725	5.907	4.970	5.913	5.757	5.314	5.088	5.419
5.125	4.670	4.952	5.500	5.432	4.452	4.439	4.575	4.488	4.968	5.170	5.892
5.683	6.056	4.842	5.697	5.834	5.818	4.742	5.803	4.839	4.906	5.863	5.505
5.766	5.080	5.125	4.439	5.006	5.547	4.942	5.348	5.170	5.477	5.536	5.337
5.716	5.880	5.418	4.836	5.094	5.612	5.337	5.698	5.456	5.251	5.602	5.980
5.012	5.347	5.220	4.959	4.927	4.730	5.385	5.433	5.950	5.588	5.492	5.285
5.819	5.650	5.389	6.025	5.607	5.882	5.749	5.699	5.507	5.914	5.102	5.224
5.505	5.680	6.097	6.088	6.007	6.021	5.487	5.107	5.015	5.422	4.956	5.836
5.904	5.287	5.472	5.419	6.003	5.021	5.678	5.532	4.687	4.666	6.026	5.509
4.574	6.046	5.694	5.650	5.683	5.961	5.823	5.719	4.948	4.840	5.128	5.322
5.578	4.830	5.955	5.595	5.133	5.714	5.254	5.227	5.006	4.955	4.931	5.250
5.715	5.424	5.606	5.870	6.029	5.842	5.801	5.011	5.300	4.909	5.343	5.767
5.092	5.543	5.202	6.062	5.637	5.365	5.429	5.304	5.333	5.154	4.950	6.002
4.867	5.155	5.084	4.996	5.433	5.808	4.669	5.294	4.908	4.748	4.602	4.495
4.439	4.839	5.916	5.971	5.696	5.810	5.073	4.863	5.438	5.784	5.679	6.059
5.944	6.081	5.997	5.902	5.223	5.671	5.732	5.765	5.568	5.367	5.848	6.017
5.171	5.842	5.950	4.954	5.494	5.459	5.243	5.946	6.064	5.442	4.980	4.547
5.126	5.361	5.400	5.459	5.202	5.195	5.079	5.173	5.828	5.779	5.886	5.353
4.729	5.104	4.485	4.979	5.229	4.778	5.861	5.835	5.700	5.205	5.567	5.822
5.955	5.914	4.752	5.401	4.860	5.701	5.432	5.610	5.566	5.334	4.976	5.902
5.496	5.189	5.808	4.851	4.793	5.154	5.449	5.054	5.176	5.226	5.610	5.278
6.004	5.294	5.727	5.128	4.989	5.151	5.397	5.770	5.134	6.065	5.340	5.730

4.894	5.112	4.439	4.439	4.900	4.803	5.888	4.923	5.049	4.822	4.989	6.053
4.805	5.232	5.164	5.855	5.512	5.941	5.727	5.612	6.090	5.766	5.245	4.975
4.825	4.786	5.245	5.205	4.513	5.252	4.806	4.547	5.226	5.029	5.950	5.515
5.624	5.278	5.223	5.844	4.625	4.750	4.795	4.936	5.725	5.596	6.007	6.035
5.090	5.992	6.030	6.048	5.269							

2008

5.641	5.613	4.750	5.428	4.917	4.578	4.522	4.502	4.677	4.508	4.502	5.012
4.511	4.689	5.233	5.072	4.632	5.302	4.810	4.798	4.603	4.702	5.134	5.334
5.152	5.139	4.966	5.356	5.257	5.330	5.744	4.618	5.225	5.765	5.284	6.001
6.099	5.425	4.720	5.791	6.023	4.932	5.116	4.899	4.951	5.190	5.358	5.427
5.197	5.563	4.925	5.980	6.088	6.022	5.982	6.103	5.471	5.133	6.042	5.186
4.695	5.953	5.371	5.443	5.273	5.226	5.880	4.578	4.637	5.106	4.789	5.311
5.988	5.727	5.321	5.823	5.013	6.050	6.073	6.093	5.420	5.687	5.431	5.145
4.711	4.855	4.720	4.654	4.617	5.257	5.720	5.925	5.908	5.355	4.794	4.631
5.244	5.220	5.943	4.540	4.537	5.796	5.355	5.256	5.685	5.564	4.824	5.622
5.963	5.013	5.434	5.379	5.307	5.397	4.942	5.513	4.485	4.614	4.604	4.668
5.993	5.341	4.959	4.834	5.028	4.997	5.065	5.153	5.721	5.440	5.848	5.185
5.207	5.115	5.438	5.381	4.743	5.745	4.924	4.993	4.439	5.646	5.130	5.094
5.000	4.718	5.232	4.928	4.833	4.646	4.878	4.754	5.035	4.439	5.574	5.189
5.183	5.640	5.390	5.593	4.978	5.881	5.620	5.383	4.765	5.372	5.561	6.097
6.040	5.504	5.131	5.131	5.300	5.708	5.877	4.783	4.783	4.579	5.176	4.744
4.744	4.781	4.814	4.630	4.439	4.710	5.443	5.068	4.568	5.158	4.476	4.651
4.810	4.804	5.234	5.219	6.074	5.734	5.312	5.176	4.736	4.589	4.887	5.464
5.065	4.713	4.528	4.614	4.985	4.568	4.452	6.007	5.680	6.100	6.059	6.094
5.956	5.735	5.713	5.080	4.742	4.804	5.266	4.570	4.550	4.833	4.961	5.585
4.495	4.992	5.210	5.652	4.603	4.880	4.752	5.174	5.271	4.827	5.030	4.879
4.924	4.886	5.263	5.484	5.992	5.748	4.899	5.534	5.049	5.655	5.520	5.564
4.720	5.177	5.544	4.953	5.555	5.632	5.350	5.090	4.958	6.006	5.592	6.086
5.611	4.860	5.378	6.083	4.881	6.042	5.799	5.986	6.028	6.009	5.919	6.096
6.100	5.346	4.439	4.439	4.795	5.689	5.973	6.051	5.184	5.072	5.845	5.935
5.623	5.626	5.469	5.672	5.787	5.376	5.092	5.178	4.936	5.840	4.784	5.354
4.895	5.104	4.776	5.118	4.949	5.861	5.409	4.959	4.463	4.463	4.509	4.873
5.683	4.915	4.715	5.954	5.965	5.989	5.444	5.577	4.776	5.606	5.175	4.539
5.316	4.884	6.076	6.064	5.898	4.439	5.643	5.904	5.624	5.188	5.068	4.553
4.439	5.456	6.072	5.420	4.971	4.839	4.870	4.809	5.096	5.620	5.406	4.929
5.340	5.996	5.368	5.377	5.515	5.764	5.252	6.080	6.090	5.072	5.962	4.914
5.500	4.928	4.923	5.250	6.075	5.193						

2009

6.051	5.594	4.795	4.439	4.439	5.282	4.796	4.736	5.891	5.291	4.966	5.453
5.632	5.505	5.913	6.033	5.948	5.475	4.661	5.169	4.439	4.896	4.939	4.911
5.285	5.803	5.773	5.766	5.048	5.957	5.922	5.569	5.912	6.054	6.055	5.148
5.119	4.819	5.768	5.908	5.550	5.193	5.908	5.606	5.962	4.975	4.729	5.855
6.075	4.439	5.436	4.729	4.993	4.439	4.485	4.465	5.479	4.792	5.574	5.065
4.439	4.439	4.609	4.783	4.864	5.242	5.681	4.883	4.486	5.843	5.821	5.546
4.835	4.494	4.544	5.445	5.556	5.645	4.683	4.492	5.567	5.469	4.824	4.962

4.442 4.724 4.862 5.223 4.972 4.589 6.084 5.847 5.291 5.955 5.867 4.745
5.815 4.565 4.727 4.535 4.847 4.559 4.625 5.012 4.764 5.499 4.920 5.430
5.226 4.821 4.690 4.569 5.212 6.045 5.794 5.725 5.493 5.429 5.573 5.916
5.881 5.609 6.011 5.582 5.884 5.338 5.782 4.994 4.708 5.239 5.919 4.993
5.961 5.931 5.545 4.814 4.650 5.062 5.385 4.823 4.439 5.030 5.651 5.248
4.634 6.027 5.974 5.088 5.818 5.284 4.727 5.624 5.493 6.080 5.887 5.253
5.330 5.730 5.736 5.993 4.594 5.009 5.533 5.140 5.415 5.526 4.894 4.630
5.849 5.167 4.901 4.889 4.906 4.726 5.124 5.006 5.338 5.969 5.573 5.936
5.578 5.850 5.231 6.071 5.293 5.243 5.570 6.052 5.523 5.087 4.501 5.418
5.593 5.456 4.738 4.703 5.991 5.923 4.486 5.385 6.082 5.040 5.477 4.733
5.785 5.956 6.013 4.716 4.496 4.681 5.231 5.493 4.937 5.207 4.734 5.366
5.160 5.182 4.865 5.033 5.019 5.291 5.556 5.008 5.637 5.738 4.681 5.741
5.351 5.175 5.082 6.057 6.103 6.017 5.379 5.413 5.671 4.496 4.698 4.857
4.611 5.903 4.444 5.551 5.644 5.856 5.465 5.708 4.511 4.597 5.053 6.072
4.609 4.741 4.896 4.455 4.548 4.801 4.696 4.571 4.716 4.439 4.439 4.439
4.554 4.574 4.627 5.180 5.739 5.417 5.987 5.444 5.945 5.692 5.875 5.194
6.066 5.280 5.243 5.556 6.057 5.524 5.089 4.489 5.395 5.602 5.471 4.721
4.680 6.016 5.927 4.474 5.454 6.084 5.036 5.506 4.719 5.797 5.928 6.012
4.703 4.501 4.674 5.243 5.493 4.902 5.209 4.746 5.334 5.162 5.185 4.856
5.045 5.039 5.247 5.507 5.026 5.566 5.724 4.693 5.751 5.332 5.153 5.103
6.044 6.101 5.999 5.350 5.443 5.607 4.489 4.681 4.833 4.643 5.920 4.451
5.541 5.675 5.837 5.479 5.708 4.527 4.589 5.015 6.068 4.599 4.757 4.917
4.461 4.551 4.790 4.700 4.570 4.710 4.439 4.439 4.439 4.561 4.568 4.614
5.173 5.720 5.458 5.986 5.481

2010

4.444 4.607 4.560 5.686 6.055 5.973 5.520 5.716 5.514 5.621 5.194 5.364
5.345 4.832 5.387 5.763 5.318 5.503 5.636 5.972 6.082 5.781 5.664 4.948
4.694 4.810 4.731 4.653 4.554 4.585 4.587 4.933 5.079 4.655 5.968 5.997
4.464 5.473 5.725 4.695 4.496 5.853 4.966 5.021 5.108 5.190 5.205 5.917
5.379 4.583 4.475 4.825 4.860 4.662 5.081 5.290 5.071 4.855 5.188 5.035
5.314 5.619 5.208 5.311 5.663 5.631 5.621 5.162 4.529 4.711 4.618 5.532
4.868 5.169 5.142 5.220 5.684 4.928 5.628 5.738 5.210 5.087 6.022 5.586
5.567 6.004 5.162 5.371 5.956 5.816 5.961 5.550 4.907 4.677 4.585 5.034
4.763 4.647 5.590 4.720 4.819 5.488 5.643 6.061 5.917 5.561 5.533 5.256
5.497 5.902 5.334 4.928 5.014 5.063 4.889 5.121 4.600 5.959 5.436 4.880
5.592 5.129 5.677 5.854 5.465 5.193 5.033 5.229 4.609 5.475 5.142 5.896
5.649 5.036 4.638 4.958 5.066 5.557 5.529 5.173 5.694 5.319 5.615 5.433
5.341 5.426 4.842 4.803 5.180 5.127 5.314 5.635 5.559 4.912 4.666 5.252
5.066 5.412 5.769 5.359 4.627 4.639 4.439 4.816 4.656 4.657 5.320 5.211
4.803 5.272 5.073 4.624 5.199 4.953 4.648 4.573 4.988 4.770 5.411 5.073
5.030 4.465 5.830 6.073 6.046 6.091 5.384 4.866 5.275 4.610 4.643 4.586
5.074 5.819 4.715 4.462 4.685 4.942 5.901 5.768 5.474 5.359 5.560 4.530
4.616 4.641 4.553 5.005 4.471 4.652 4.833 4.943 4.940 4.984 5.368 5.138
4.644 5.160 5.489 4.535 4.802 5.361 5.816 5.769 5.105 5.467 4.963 5.716
4.535 4.511 4.946 5.887 5.618 5.429 5.418 4.641 4.992 4.983 4.478 5.936
5.832 5.860 5.166 5.604 5.494 5.172 5.928 4.451 4.558 4.631 4.526 5.906

5.767	5.949	5.075	5.703	5.370	4.439	5.042	4.540	4.950	4.923	4.439	4.660
4.749	5.641	5.143	5.324	5.343	4.803	4.567	4.915	5.060	5.150	4.967	5.637
5.871	5.253	5.399	5.939	5.798	5.365	5.467	4.700	4.712	4.830	4.439	4.439
4.521	4.990	5.396	4.569	5.273	5.785	5.834	5.330	5.905	6.035	5.674	6.022
5.526	6.062	6.025	5.674	5.731	5.934	5.216	5.134	4.699	4.645	4.594	4.729
4.792	5.087	4.532	4.902	4.743	5.078	5.327	5.745	5.083	4.622	4.744	4.820
4.439	4.696	5.795	5.572	5.641	5.548	5.458	5.051	5.183	5.984	5.885	5.000
5.841	4.957	5.023	5.591	4.664	4.709	4.782	5.639	5.033	4.565	5.155	5.004
4.951	5.932	5.954	5.444	5.293	5.134	5.013	5.661	5.660	5.066	5.106	5.122
5.129	5.426	5.991	5.189	5.034							

6.1. Crises Pattern

Let us begin with the following observations. The values of entropy for 7 days, prior to 04.03.2001 and 22.09.2008, i.e. 5 and 7 days before big crises, are presented in Table 6. Values in the same columns of table 7 are significantly different. Both sequences are marked by gray background in Table 6 in the places, corresponding to these dates.

Entropy values prior crises

Table 7

Day	26.02.01	27.02.01	28.02.01	01.03.01	02.03.01	03.03.01	04.03.01
Entropy	4.887	5.854	5.981	5.757	6.028	5.351	5.175
Day	16.09.08	17.09.08	18.09.08	19.09.08	20.09.08	21.09.08	22.09.08
Entropy	5.090	4.958	6.006	5.592	6.086	5.611	4.860

Denote values in the 1st column as x_1 and y_1, in the 2nd column as x_2 и y_2, and so on, till to values in the 7th column, denoted as x_7 и y_7. The values $x_1, x_2, \ldots, x_7$ and values $y_1, y_2, \ldots, y_7$ satisfy the following system of inequalities:

$$\begin{cases} z_5 > z_1, z_5 > z_2, z_5 > z_3, z_5 > z_4, z_5 > z_6, z_5 > z_7, \\ z_3 > z_1, z_3 > z_2, z_3 > z_4, z_3 > z_6, z_2 > z_7, \\ z_6 > z_7, z_5 > 6, z_3 > 5.7, z_4 < 6. \end{cases} \tag{11}$$

Inequalities from the 1st row mean that value z_5 is greater than all the other values; inequalities from the 2nd row mean that value z_3 is greater than all the other values, except z_5; next inequality $z_6 > z_7$ (together with inequality $z_5 > z_6$) means that three last values monotonously decrease. The three last inequalities express one-sided constraints of values z_5, z_3 and z_4.

Consecutive 7 values of entropy whose 7^{th} value correspond to arbitrary day x can satisfy or not satisfy to system of linear inequalities (11). We see that 7-tuples that correspond to 04.03.2001 and to 22.09.2008 satisfy to system (11). These 7-tuples are marked be gray background in Table 6. Moreover, there are only three days during all the 10-year period, whose 7-tuple satisfy system (8) – except the two above-mentioned cases. These cases are also marked in Table 6. Therefore, it is possible to consider system (11) as a ***pattern***, which corresponds to beginning of big crises.

Only three times for 3652 days (10 years) the found pattern became a phantom – its prediction of crisis was wrong, and no crisis happened. Two times, it gave the correct prediction of the big crises. At the same time, the absence of the pattern gives 100% guaranty of the absence of crisis in the next several days. It is possible to suppose that this knowledge can help to stock market specialists in elaboration of reasonable trading strategy (see, for instance, the article [Aleskerov and Egorova, 2012]). However, these topics require a special attention; they will be considered in future investigations.

Very appreciable book [Reinhart and Rogoff, 2010] states that all the crises at stock markets have many common prior markers. It is possible to say that the result of Section 6 is one of the formal expressions of this general assertion.

I would like to add that the literature, devoted to attempts of crises prediction, is very abundant and diversified. Any reasonable review goes far beyond the framework of the presented material, whose main goal consists in presentation of the new decomposition approach to data analysis. The material from three last sections 4 – 6 demonstrates only the possibility of application of the suggested approach in various real situations.

7. Conclusion

Let us give some final remarks and comments to the above presented material.

1. There are a few parameters in the essential algorithm of family of graph decompositions construction: two parameters in the algorithm of graph dichotomy, one parameter of consecutive dichotomies and two parameters of DAA repetitions. The choice of these parameters has not accompanied by any explanations. It is possible to say that the suggested

algorithm is the algorithm of data analysis and it is not the algorithm of a system behavior imitation. The only requirement to the algorithm itself and to its parameters consists in usefulness of this algorithm.

2. The applied results from Sections 4, 5 and 6 required different versions of the suggested general scheme of the essential algorithm application. Another time it must be pointed to the necessity of manual parameters selection. Unfortunately, this is done by trials and errors method. It is supposed to improve it in future investigations.

3. It seems that additional use of some other parameters can get rid out of very few superfluous non-crisis days with crisis pattern (phantoms). It will be done in future investigations.

4. Some elements of the general scheme can be improved. Particularly, in formulae (5) and (7) it is possible to define value of parameter s in dependence of convergence of the corresponding averages of index values (adaptively). This version can significantly reduce calculation time.

5. All the experimental results concern only one parliament and only one stock market. Of course, in the further investigation it is supposed to consider essentially wider data sets. It will be done as the required information becomes available.

6. The last remark is as follows. Several times in Section 6 the expression "big crisis" is mentioned without any formal definition. It seems that to give a formal definition of this notion is practically impossible. However, such a definition is unessential. We can simply suppose that a big cri-sis is a ***state of the stock market*, *which the most of participants perceive as a big crisis***. And this assumption leads to their behavior, characterized by the found pattern.

The author is grateful to F.T. Aleskerov for his support and attention to the work, B.G. Mirkin and G.I. Penikas for helpful discussions, V.I. Jakuba for help in mastering various computing shells, and E.V. Misharina for S&P-500 data extraction.

The article was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and supported within the framework of a subsidy by the Russian Academic Excellence Project '5-100'.

References

Aleskerov F., Egorova L. Is it so bad that we cannot recognize black swans? // Economics Letters. 2012. Vol. 117. No. 3. P. 563–565.

Boginski V., Butenko S., Pardalos P.M. Statistical analysis of financial networks, Compu-tational Statistics & Data Analysis. 2005. No. 48. P. 431–443.

Feller W. An Introduction to Probability Theory and Its Applications. Revised Printing. John Wiley & Sons. 1970. (См. также русское издание: Феллер В. Введение в теорию вероятностей и ее приложения. М.: Мир, 1984. Т. 1.)

Girvan M., Newman M.E.J. Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 7821–7826, 2002.

Goodman S.E., Hedetniemi S.T. Introduction to the design and analysis of algorithms. Mc-Grow-Hill Book Company, 1977. (См. также русское издание: Гудман С., Хидетниеми С. Введение в анализ и разработку алгоритмов. М.: Мир, 1981.)

Luxburg U. A Tutorial on Spectral Clustering // Statistics and Computing. 2007. Vol. 17. No. 4. P. 395–416.

McCabe T.J. A complexity measure // IEEE Transactions on software Engineering. 1976. No. 4. P. 308–320.

Mirkin, B. Clustering: A Data Recovery Approach. Chapman & Hall/CRC, 2012.

Mirkin B. Core Concepts in Data Analysis: Summarization, Correlation, Visualization. Springer, 2011.

Pincus S. M. Approximate entropy as a measure of system complexity // Proceedings of the National Academy of Sciences. 1991. Vol. 88. No. 6. P. 2297–2301.

Raychev N. Measure of entanglement by Singular Value decomposetion // International Journal of Scientific and Engineering Research. 2015. Vol. 6. No. 7. P. 1350–1355.

Reinhart C., Rogoff K. This time is different. Princeton University Press, 2009. (См. также русское издание: Рейнхарт К, Рогофф К. На этот раз все будет иначе. М.: Карьера Пресс, 2011.)

Rubchinsky A. Divisive-Agglomerative Algorithm and Complexity of Automatic Classification Problems: Working Paper NRU HSE WP7/ 2015/09. <www.hse.ru/en/org/ hse/wp/wp7en>

Rubchinsky A. Divisive-Agglomerative Classification Algorithm Based on the Minimax Modification of Frequency Approach: Working Paper NRU HSE WP7/2010/07. Moscow, 2010. <www.hse.ru/en/org/hse/wp/wp7en>

Алескеров Ф.Т. и др. (In Russian). Влияние и структурная устойчивость в Российском парламенте (1905–1917 и 1993–2005 гг.). М.: ФИЗМАТЛИТ, 2007.

Рубчинский, А. А.

Семейство декомпозиций графа и его применения для анализа данных [Текст] : препринт WP7/2016/09 / А. А. Рубчинский ; Нац. исслед. ун-т «Высшая школа экономики». – М. : Изд. дом Высшей школы экономики, 2016. – (Серия WP7 «Математические методы анализа решений в экономике, бизнесе и политике»). – 60 с. – 35 экз. (На англ. яз.)

Предложен новый классификационный подход к анализу сложных систем. Традиционный подход связан с построением одной, «наиболее правильной», декомпозиции рассматриваемой системы. В то же время предложенный подход ориентирован на построение семейства классификаций, чьи свойства позволяют выявить некоторые важные содержательные особенности исходной системы.

Целесообразность и применимость разработанного подхода проиллюстрированы тремя хорошо известными и важными примерами: автоматической классификацией, голосованием в политических органах и фондовым рынком. В рассмотренных случаях представленные результаты не могут быть получены известными методами. Это подтверждает преимущества предложенного подхода.

Ключевые слова: декомпозиция графов; автоматическая классификация; показатель сложности; энтропия; голосование в политических органах; фондовый рынок; кризис

Рубчинский А.А., Национальный исследовательский университет «Высшая школа экономики», Международная научно-учебная лаборатория анализа и выбора решений; Национальный исследовательский технологический университет «МИСИС», кафедры инженерной кибернетики и бизнес-информатики.

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

Препринт WP7/2016/09
Серия WP7
Математические методы анализа решений
в экономике, бизнесе и политике

Рубчинский Александр Анатольевич

Семейство декомпозиций графа
и его применения для анализа данных

(*на английском языке*)

Зав. редакцией оперативного выпуска *А.В. Заиченко*
Технический редактор *Ю.Н. Петрина*

Отпечатано в типографии
Национального исследовательского университета
«Высшая школа экономики» с представленного оригинал-макета
Формат 60×84 $^{1}/_{16}$. Тираж 35 экз. Уч.-изд. л. 3,6
Усл. печ. л. 3,5. Заказ № . Изд. № 1993

Национальный исследовательский университет
«Высшая школа экономики»
125319, Москва, Кочновский проезд, 3
Типография Национального исследовательского университета
«Высшая школа экономики»