http://doi.org/10.35784/iapgos.2591

DESCRIPTION OF ALGORITHMS FOR BALANCING NUMERICAL MATRICES AND THEIR DIVISION INTO HIERARCHICAL LEVELS ACCORDING TO THEIR TYPE AND COMPLEXITY

Yuriy Khanas[1], Michał Borecki[2]
[1]Lviv Polytechnic National University, Lviv, Ukraine, [2]Warsaw University of Technology, Warsaw, Poland

Abstract. *This article describes a set of algorithms for so-called balancing of numerical matrices, which were developed by the author. Each section consists of several algorithms that are divided into different levels. The order of these levels depended on the chronology of the creation of certain algorithms. Chronology also affected the complexity of these balancing algorithms, so it can be argued that the algorithms are described in order from the simplest level to the most complex. It is important to emphasize that the purpose of the article is to describe the actions on matrices that determine the balancing algorithm of a certain level, and practical application will be the next step.*

Keywords: matrix balancing, separated matrix sectors, solid matrix sector, virtual matrix boundary, non-uniform matrix

ALGORYTMY BILANSOWANIA ORAZ HIERARCHIZACJI MACIERZY WEDŁUG ICH TYPU I ZŁOŻONOŚCI

Streszczenie. *Artykuł przedstawia autorskie algorytmy bilansowania macierzy. Każdy rozdział składa się z kilku algorytmów, które są rozdzielone na różne poziomy. Te poziomy są uporządkowane w zależności od chronologii ich stworzenia. Podobnie chronologia ma wpływ na złożoność algorytmów zbilansowania, w związku z tym można stwierdzać, że algorytmy są uszeregowana według stopnia złożoności. Niniejszy artykuł jest pierwszym etapem pokazującym sposób zbilansowania pewnego poziomu macierzy, natomiast kolejnym etapem będzie efekt praktyczny.*

Słowa kluczowe: bilansowanie macierzy, rozdzielone sektory macierzy, cały sektor macierzy, granica wirtualna macierzy, wyznaczona granica macierzy

Introduction

Some of the described algorithms have been mentioned earlier in previous articles of the author [1–7], but some names, designations, definitions, steps and actions may be partially or completely different, due to constant research and development of this "family" of algorithms.

The very first steps in the development of the author were aimed at creating algorithms for data compression [8], initially experiments were performed with text compression. In the further work manipulations with alphabets, keys, the size of the text and other aspects which influenced formation of a numerical matrix were carried out. After that, the obtained matrices of numbers were changed by various manipulations, the size of the matrix and the content were changed. Thus, the author came to the possibility of reducing numerical matrices [9–13]. At the beginning of matrix reduction studies, the results of each iteration had quite unpredictable consequences for the size of these matrices, for example, the number of columns could only decrease, and the number of rows could both decrease and increase. Then the author focused on methodological support, as a result of which various rules, remarks and something like theorems were developed. As experiments on matrix reduction options continued, several interesting cases arose – namely, matrices that could be completely reduced and disappeared. Matrices that were not subject to reduction were also identified and it was impossible to perform

at least one iteration. At first glance, this was a problem because such a matrix could not be reduced or restored, it was then that the author developed so-called "auxiliary" algorithms that allowed to change the contents of the matrix so that at least one iteration of reduction was allowed, but it was not reduced completely. All developments and modifications of further algorithms were accompanied by methodological support, which in turn grew more and more. Eventually, a compression algorithm was developed to reduce the amount of text data and return it to its original form.

Testing of the already developed compression method showed quite different results. In some cases, the new algorithm was more efficient than existing counterparts, and sometimes less efficient. Even the most unsuccessful experiments did not disappoint the author, but on the contrary prompted to invent a new approach, algorithm, rule, exclusion from the rules or a new key system for compressing and restoring text.

The author's plans were not limited to test information, it was the turn of research on image compression algorithms, signals, and even the development of compression algorithms with data encryption. However, in the study of data encryption algorithms, the idea arose to apply this method to encryption, and postpone compression for later.

When adapting the algorithm for data encryption, there was no need to reduce the matrices, then it was decided to take as a basis auxiliary algorithms, which the author invented earlier. The first experiments showed quite good results, even most of the methodological support remained relevant. In fact, then the algorithm moved in a new direction and developed rapidly. Through modifications, finding possible problems, the number of algorithms has grown. In fact, then there was a need to systematize them so that it was possible to clearly present the method of encryption in publications and other scientific papers.

The first sorting of all variants of algorithms had the following structure: basic algorithms, auxiliary, modified, combined, additional. However, it is difficult to lay distinguish between modified and combined algorithms, as between the subsidiary and additional, as are algorithms that can belong to multiple categories.

To avoid confusion in further work on the development and improvement of this method, the author proposed to divide these algorithms into levels, which may include several balancing algorithms.

Although this algorithm is effectively used to encrypt text, it is not a limitation for it, because at this stage several areas of science have been found where its application has the right to life.

Once again, I would like to note that in this article the emphasis is on the description of actions on matrices actually balancing algorithms. Some auxiliary algorithms have already been mentioned by the author in previous articles, such as partitioning and reflection of the matrix [11,12]. Here the focus is on balancing and dividing it into different levels, each of which will be described in its own section of the article. This article is an introduction and the basis for subsequent new articles to gradually introduce fellow scientists and other readers of this work into the field of its development, without confusing anyone.

The author emphasizes that each individual algorithm and method as a whole is constantly being developed, modified, improved, constant research is conducted, but any changes will be described in subsequent articles.

It can also be noted that the method is a kind of innovation of the author (so far in the field of data encryption), respectively, the actions, steps and manipulations shown may or may not coincide with some algorithms and mathematical operations.

1. First level balancing algorithms

The first level includes two algorithms:

- horizontal balancing,
- vertical balancing.

In the initial stages of research, they were called basic algorithms because they are the basis for the algorithms that will be described below, i.e. all modifications and combinations originate from them.

In this article, these algorithms will have a slightly more complex description and additional definitions that will allow you to more clearly see the difference between these levels. These definitions will be used in future works [12–22].

Each of the balancing types is designed for certain proportions of the matrix. Horizontal balancing is designed for a matrix that has an even number of columns and an odd number of rows. Vertical balancing, in turn, is designed for matrices with an odd number of columns and an even number of rows.

Consider an example with matrices for each of the cases, the matrices are formed from completely random numbers.

$$\underset{\textbf{A}}{\begin{array}{cccc} 2 & 1 & 7 & 4 \\ 4 & 7 & 7 & 0 \\ 1 & 8 & 4 & 3 \end{array}} \qquad \underset{\textbf{B}}{\begin{array}{ccc} 4 & 0 & 3 \\ 7 & 7 & 4 \\ 1 & 7 & 8 \\ 2 & 4 & 1 \end{array}}$$

Fig. 1. Starting matrices for balancing the first level

Two simple matrices for convenient and clear presentation of algorithm steps.

Accordingly, Fig. 1a – for horizontal balancing, Fig. 1b – for vertical balancing.

To demonstrate how each balancing occurs, each matrix should be divided into two equal parts, i.e. so that each part has the same number of digits. The so-called "virtual boundary" will be used for this purpose (in earlier works it was called simply the center of a matrix).

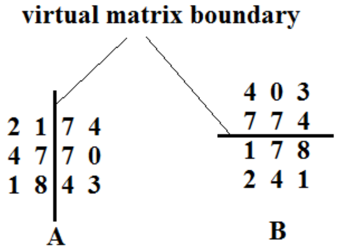


Fig. 2. Separation of matrices by a virtual boundary

When the matrices are separated by a boundary (center), then the opposite parts of the matrix are compared. In the matrix for horizontal balancing (Fig. 2a) the opposite rows will be compared, and in the matrix for vertical balancing the opposite columns will be compared. The sum of the numbers of a row or column separated by a boundary will be compared. This is called a sector-by-sector comparison.

$$\underset{\textbf{A}}{\begin{array}{cccc|ccc} + & (3) & 2 & 1 & 7 & 4 & (11)\ - \\ - & (11) & 4 & 7 & 7 & 0 & (7)\ + \\ - & (9) & 1 & 8 & 4 & 3 & (7)\ - \end{array}} \qquad \underset{\textbf{B}}{\begin{array}{ccc} - & + & + \\ (11) & (7) & (7) \\ 4 & 0 & 3 \\ 7 & 7 & 4 \\ \hline 1 & 7 & 8 \\ 2 & 4 & 1 \\ + & - & - \end{array}}$$

Fig. 3. Comparison of individual sectors of the matrix

The results of the comparison of sectors are shown in the figure, in parentheses are the sums of opposite rows and columns. Accordingly, the row or column with the larger sum should be reduced, and the sector with the smaller sum should be increased. This operation is as follows – when you increase the sector, all digits (row or column) increase by one, and when you decrease all the digits of the sector are reduced by one. The figure also shows the actions of increase and decrease (plus and minus).

Now we will iterate the balancing of the first level on both matrices.

$$\underset{\textbf{A}}{\begin{array}{cc|cc} 3 & 2 & 6 & 3 \\ 3 & 6 & 8 & 1 \\ 0 & 7 & 5 & 4 \end{array}} \qquad \underset{\textbf{B}}{\begin{array}{ccc} 3 & 1 & 4 \\ 6 & 8 & 5 \\ \hline 2 & 6 & 7 \\ 3 & 3 & 0 \end{array}}$$

Fig. 4. Matrices after balancing the first level

Since matrices with the same numbers were chosen for both balances, the result is the same in both. At first glance, this is a primitive algorithm with simple actions and steps, but we should not forget that this is only the first algorithm and it is designed to provide an accessible explanation of the logic of the developed method.

Some publications have already described the remark about the cyclic change of matrix numbers, but it will be useful to recall it. The matrix uses unique, positive, integers (0–9), so to preserve the integrity of the matrix, the author introduced a rule: if when reducing the sector there is a number zero, then after reducing it by one, zero will become the number nine, and vice versa, if the number nine is present when the sector increases, then it becomes zero. That is, the numbers will change cyclically in the range 0–9.

However, the author has developed separate algorithms for working with fractional, multi-digit and negative numbers, but they have applications in another area, so they will not be described in this article.

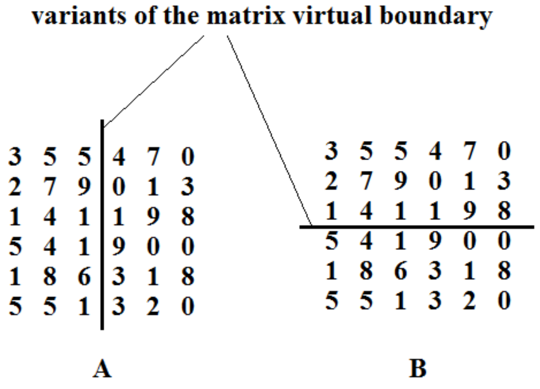
There are also cases when the matrix has an even number of rows and columns, then depending on their number you can determine which type of balancing will be optimal. And if the matrix has an even number of rows and columns and their number is the same. In this case, the user can choose the type of balancing at random, or conduct each balancing separately and compare the results, and then based on them to make your choice.

Consider ode from such examples using a small matrix. Hereinafter, such matrices will be called paired symmetric (uniform) matrices.

$$\begin{array}{cccccc} 3 & 5 & 5 & 4 & 7 & 0 \\ 2 & 7 & 9 & 0 & 1 & 3 \\ 1 & 4 & 1 & 1 & 9 & 8 \\ 5 & 4 & 1 & 9 & 0 & 0 \\ 1 & 8 & 6 & 3 & 1 & 8 \\ 5 & 5 & 1 & 3 & 2 & 0 \end{array}$$

Fig. 5. Paired symmetric matrix

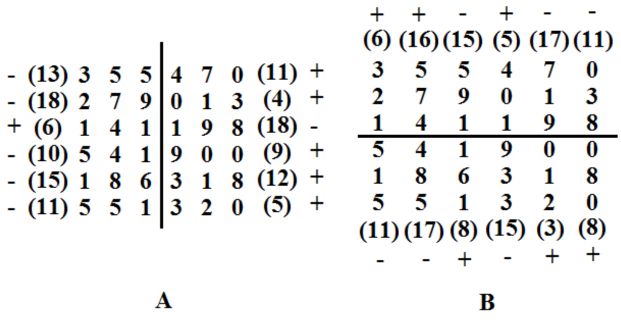
Now we need to give a virtual limit, and in this example we give two – one for each type of balancing, and then compare the results.



$$A:\ \begin{array}{ccc|ccc} 3&5&5&4&7&0\\ 2&7&9&0&1&3\\ 1&4&1&1&9&8\\ 5&4&1&9&0&0\\ 1&8&6&3&1&8\\ 5&5&1&3&2&0 \end{array}\qquad B:\ \begin{array}{cccccc} 3&5&5&4&7&0\\ 2&7&9&0&1&3\\ 1&4&1&1&9&8\\ \hline 5&4&1&9&0&0\\ 1&8&6&3&1&8\\ 5&5&1&3&2&0 \end{array}$$

Fig. 6. Virtual boundaries of the matrix for balancing both types

The principle remains the same, comparing by divided sectors, determining the increase and decrease for each of the sectors.

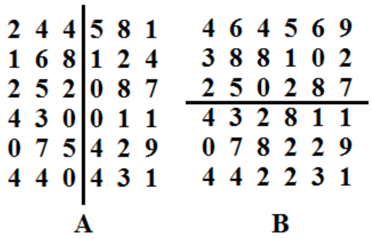


$$A:\ \begin{array}{cc|ccc|ccc|cc} - & (13) & 3&5&5&4&7&0&(11)&+\\ - & (18) & 2&7&9&0&1&3&(4)&+\\ + & (6) & 1&4&1&1&9&8&(18)&-\\ - & (10) & 5&4&1&9&0&0&(9)&+\\ - & (15) & 1&8&6&3&1&8&(12)&+\\ - & (11) & 5&5&1&3&2&0&(5)&+ \end{array}\qquad B:\ \begin{array}{cccccc} + & + & - & + & - & -\\ (6)&(16)&(15)&(5)&(17)&(11)\\ 3&5&5&4&7&0\\ 2&7&9&0&1&3\\ 1&4&1&1&9&8\\ \hline 5&4&1&9&0&0\\ 1&8&6&3&1&8\\ 5&5&1&3&2&0\\ (11)&(17)&(8)&(15)&(3)&(8)\\ - & - & + & - & + & + \end{array}$$

Fig. 7. Comparison of individual sectors of the matrix

Even after comparing the sectors, it can be seen that the number of sectors for increasing and decreasing differs depending on the boundary of the matrix, ie it is immediately apparent that the results of horizontal and vertical balancing will be different from each other, despite the fact that they are the same matrix.

Let's carry out one iteration of each balancing.



$$A:\ \begin{array}{ccc|ccc} 2&4&4&5&8&1\\ 1&6&8&1&2&4\\ 2&5&2&0&8&7\\ 4&3&0&0&1&1\\ 0&7&5&4&2&9\\ 4&4&0&4&3&1 \end{array}\qquad B:\ \begin{array}{cccccc} 4&6&4&5&6&9\\ 3&8&8&1&0&2\\ 2&5&0&2&8&7\\ \hline 4&3&2&8&1&1\\ 0&7&8&2&2&9\\ 4&4&2&2&3&1 \end{array}$$

Fig. 8. The results of balancing the matrix

In general, the results of balancing the matrix differ, although some balanced numbers are the same in both cases. This fact does not indicate the ineffectiveness of any type of balancing. Each of the two types is effective, but to explore it in more detail, it would be necessary to compare not only numerical matrices, but also, for example, encrypted texts. Then you can judge which of the types of balancing will be optimal for this example. For different texts or other data, or even other scientific problems, any of the algorithms described in the article may be the most effective.

2. Second level balancing algorithms

Among the algorithms of the second level there are also our old acquaintances, namely horizontal and vertical balancing, but with a small difference – in the previous section there was a virtual boundary that divided the matrix, and at this level we have a certain limit of matrix division.

The defined boundary (center) of the matrix is the central row or column of the matrix, which will not change (its numbers will be neither increased nor decreased, but will simply remain unchanged).

The question arises – for which matrices is the balance of the second level? The answer is quite obvious: balancing of this level is intended for matrices with an odd number of rows and columns.

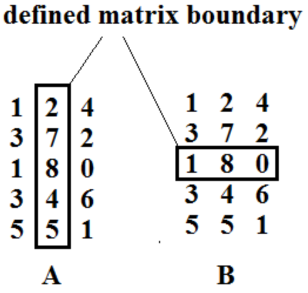
This section will cover a number of examples, namely examples with different ratios of an odd number of rows and columns, with both types of balancing applied to each example to illustrate the results. Also, both types of balancing will be used for a matrix with an odd number and the same number of rows and columns. Such matrices will hereinafter be called odd symmetric (uniform) matrices.

For the first example, a matrix is chosen where the number of rows is greater than the number of columns (and their number is odd).

$$\begin{array}{ccc} 1&2&4\\ 3&7&2\\ 1&8&0\\ 3&4&6\\ 5&5&1 \end{array}$$

Fig. 9. Input matrix for balancing

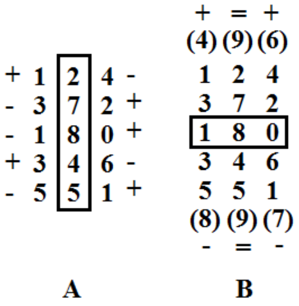
This matrix has 5 rows and 3 columns, in both cases we can select the border in the form of a central row or a central column of the matrix. Therefore, we select both boundaries of the matrix and perform both types of balancing.



$$A:\ \begin{array}{c|c|c} 1&2&4\\ 3&7&2\\ 1&8&0\\ 3&4&6\\ 5&5&1 \end{array}\qquad B:\ \begin{array}{ccc} 1&2&4\\ 3&7&2\\ \hline 1&8&0\\ \hline 3&4&6\\ 5&5&1 \end{array}$$

Fig. 10. Matrix with defined boundaries for horizontal and vertical balancing

Next, the standard steps for the algorithms are performed, namely the comparison of opposite sectors and the definition of sectors to increase and decrease. Previously, this process was called determining the direction of balancing, but this definition will be more logical.



$$A:\ \begin{array}{cc|c|cc} + & 1 & 2 & 4 & -\\ - & 3 & 7 & 2 & +\\ - & 1 & 8 & 0 & +\\ + & 3 & 4 & 6 & -\\ - & 5 & 5 & 1 & + \end{array}\qquad B:\ \begin{array}{ccc} + & = & +\\ (4)&(9)&(6)\\ 1&2&4\\ 3&7&2\\ \hline 1&8&0\\ \hline 3&4&6\\ 5&5&1\\ (8)&(9)&(7)\\ - & = & - \end{array}$$

Fig. 11. Comparison of sectors to increase and decrease them

In the first version (Fig. 11a). For horizontal balancing the sum of numbers in the sectors were not written, because in each of the opposite sectors there is only one digit. In the second variant (Fig. 11b). There is a more interesting case – there are two opposite sectors of the matrix with the same sums, they are denoted by an equal sign. In some previous articles, such cases also occurred, then the user could choose which of the sectors to increase and which to decrease. In this case, we will leave these sectors unchanged. This example is worth remembering, as it will be involved in next-level algorithms, where this kind of situation will be relevant and will be resolved.

$$\begin{array}{cc}
\begin{array}{c|c|c} 2&2&3\\ 2&7&3\\ 0&8&1\\ 4&4&6\\ 4&5&2 \end{array} &
\begin{array}{ccc} 2&2&5\\ 4&7&3\\ \hline 1&8&0\\ \hline 2&5&5\\ 4&9&2 \end{array}\\
\textbf{A} & \textbf{B}
\end{array}$$

Fig. 12. Balancing of odd non-uniform matrices

For this rather simple example, the first option (horizontal balancing) looks more effective than the second (vertical balancing). This conclusion can be reached due to a larger number of balanced numbers. Because the second option had two sectors with the same sums of numbers, they were not balanced. Such cases will not happen more than once and for each case the author has developed modifications of algorithms, which will be shown in higher-level algorithms.

Now let's try this level algorithm for a matrix with more columns and fewer rows (in both cases their number will be odd).

$$\begin{array}{ccccc} 7&5&6&2&3\\ 2&2&1&9&0\\ 3&4&7&0&8 \end{array}$$

Fig. 13. Input matrix for balancing

An odd asymmetric matrix is obtained again, except that this time we have 5 columns and three rows.

We will draw certain boundaries for horizontal and vertical balancing.

$$\begin{array}{cc}
\begin{array}{cc|c|cc} 7&5&6&2&3\\ 2&2&1&9&0\\ 3&4&7&0&8 \end{array} &
\begin{array}{ccccc} 7&5&6&2&3\\ \hline 2&2&1&9&0\\ \hline 3&4&7&0&8 \end{array}\\
\textbf{A} & \textbf{B}
\end{array}$$

Fig. 14. Defining the boundaries of the matrix

Now we redefine the sums of the numbers in the opposite sectors of the matrix, then increase and decrease the corresponding sectors. Let's not forget that a certain limit is not subject to balancing, so it will remain unchanged.

$$\begin{array}{cc}
\begin{array}{cccc|c|cccc} -&(12)&7&5&6&2&3&(5)&+\\ +&(4)&2&2&1&9&0&(9)&-\\ +&(7)&3&4&7&0&8&(8)&- \end{array} &
\begin{array}{ccccc} -&-&+&-&+\\ 7&5&6&2&3\\ \hline 2&2&1&9&0\\ \hline 3&4&7&0&8\\ +&+&-&+&- \end{array}\\
\textbf{A} & \textbf{B}
\end{array}$$

Fig. 15. The results of comparing opposite sectors

This time there are no opposite sectors with the same sums of numbers, so all sectors will be balanced. In the variant for vertical balancing (Fig. 14b.), The amounts did not fit, as each opposite sector consists of only one digit.

$$\begin{array}{cc}
\begin{array}{ccccc} 6&4&6&3&4\\ 4&4&1&0&9\\ 4&5&7&9&7 \end{array} &
\begin{array}{ccccc} 6&4&7&1&4\\ 2&2&1&9&0\\ 4&5&6&1&7 \end{array}\\
\textbf{A} & \textbf{B}
\end{array}$$

Fig. 16. Horizontally and vertically balanced matrix

As a result, both versions of the matrix were balanced without any difficulty. Of course, some figures in the balanced sectors turned out to be the same in both cases, but in general the result is different. This is due to the small size of the matrix, but in such a simple example it is easier to explain all the steps in an accessible way.

The last example in this section is the balancing of an odd uniform (symmetric) matrix in both ways.

For example, take a matrix of size 5x5.

$$\begin{array}{ccccc} 3&1&8&3&2\\ 4&3&2&7&4\\ 1&3&2&2&8\\ 6&1&0&7&7\\ 1&3&5&7&9 \end{array}$$

Fig. 17. Uniform odd matrix

This matrix will also be balanced horizontally and vertically, after which we compare the results.

So let's start with the defined boundaries of the matrix.

$$\begin{array}{cc}
\begin{array}{cc|c|cc} 3&1&8&3&2\\ 4&3&2&7&4\\ 1&3&2&2&8\\ 6&1&0&7&7\\ 1&3&5&7&9 \end{array} &
\begin{array}{ccccc} 3&1&8&3&2\\ 4&3&2&7&4\\ \hline 1&3&2&2&8\\ \hline 6&1&0&7&7\\ 1&3&5&7&9 \end{array}\\
\textbf{A} & \textbf{B}
\end{array}$$

Fig. 18. The limits of the matrix for the two balancing options are defined

Next, we again compare the sectors of the matrix with each other, write the sums of the numbers of sectors and indicate which of the sectors will be increased and which will be reduced.

$$\begin{array}{cc}
\begin{array}{cccc|c|cccc} +&(4)&3&1&8&3&2&(5)&-\\ +&(7)&4&3&2&7&4&(11)&-\\ +&(4)&1&3&2&2&8&(10)&-\\ +&(7)&6&1&0&7&7&(14)&-\\ +&(4)&1&3&5&7&9&(16)&- \end{array} &
\begin{array}{ccccc} =&=&-&+&+\\ (7)&(4)&(10)&(10)&(6)\\ 3&1&8&3&2\\ 4&3&2&7&4\\ \hline 1&3&2&2&8\\ \hline 6&1&0&7&7\\ 1&3&5&7&9\\ (7)&(4)&(5)&(14)&(16)\\ =&=&+&-&- \end{array}\\
\textbf{A} & \textbf{B}
\end{array}$$

Fig. 19. Comparison of sectors of an odd symmetric matrix

In the matrix for horizontal balancing (Fig. 18a.) It is seen that all sectors on the left will be increased, and all sectors on the right will be reduced. Such cases may occur from time to time, but they do not constitute any obstacle to obtaining correct results.

Instead, the matrix for vertical balancing has four sectors (two pairs of opposite sectors) that are equal, ie the sum of the numbers of these sectors is the same. We will not balance such sectors (as in the previous similar example), ie they will remain unchanged, as well as the defined limit of the matrix.

$$\underset{\textbf{A}}{\begin{array}{cc|c|cc} 4 & 2 & 8 & 2 & 1 \\ 5 & 4 & 2 & 6 & 3 \\ 2 & 4 & 2 & 1 & 7 \\ 7 & 1 & 0 & 6 & 6 \\ 2 & 4 & 5 & 6 & 8 \end{array}} \qquad \underset{\textbf{B}}{\begin{array}{ccccc} 3 & 1 & 7 & 4 & 3 \\ 4 & 3 & 1 & 8 & 5 \\ \hline 1 & 3 & 2 & 2 & 8 \\ \hline 6 & 1 & 1 & 6 & 6 \\ 1 & 3 & 6 & 6 & 8 \end{array}}$$

Fig. 20. The results of balancing an odd uniform matrix

It is immediately clear that horizontal balancing for this variant of the matrix is more efficient than vertical, due to the sectors of the matrix, which were equal in vertical balancing. Therefore, in the first case, all the numbers were balanced except for a certain limit. Although when using combined types of balancing, one of the algorithms used may give an ineffective result, but if we take into account the total results of all involved algorithms, then the applied method will give high results (this is tested in encryption problems).

3. Third level balancing algorithms

The third level balancing algorithms contain some steps from the previous levels and several features of their modification. Here, too, the balancing algorithm is performed horizontally and vertically. In addition, both the virtual boundary of the matrix and the defined one will be taken into account. The peculiarity of the third level balancing is that the separated sectors of the matrix will no longer be compared, but continuous opposite sectors. This means that the adjacent rows or columns will not be compared, but all the digits of both halves of the matrix separated by a border (center). In this way, there will be no more cases when several separate sectors will be equal to each other. The algorithm can be used for matrices of any proportions: even, odd, symmetric, non-uniform matrices. That is, the number of rows or columns will not limit the capabilities of the algorithm.

The most obvious example is a paired symmetric matrix, which will be balanced horizontally and vertically, after which it will be possible to compare the results.

$$\underset{\textbf{A}}{\begin{array}{ccc|ccc} 1 & 8 & 4 & 6 & 4 & 0 \\ 3 & 1 & 2 & 4 & 0 & 1 \\ 6 & 7 & 8 & 3 & 3 & 3 \\ 2 & 3 & 1 & 6 & 2 & 2 \\ 7 & 4 & 5 & 9 & 1 & 0 \\ 4 & 1 & 9 & 0 & 9 & 1 \end{array}} \qquad \underset{\textbf{B}}{\begin{array}{cccccc} 1 & 8 & 4 & 6 & 4 & 0 \\ 3 & 1 & 2 & 4 & 0 & 1 \\ 6 & 7 & 8 & 3 & 3 & 3 \\ \hline 2 & 3 & 1 & 6 & 2 & 2 \\ 7 & 4 & 5 & 9 & 1 & 0 \\ 4 & 1 & 9 & 0 & 9 & 1 \end{array}}$$

Fig. 21. Location of the matrix

Unlike the previous level algorithms, all numbers of the matrix sector on each side of the boundary will be added. It is assumed that absolutely all numbers on each side of the matrix will be counted as one continuous sector. This can be seen in more detail in the example.

$$\underset{\textbf{A}}{-\ (76) \begin{array}{|ccc|ccc|} \hline 1 & 8 & 4 & 6 & 4 & 0 \\ 3 & 1 & 2 & 4 & 0 & 1 \\ 6 & 7 & 8 & 3 & 3 & 3 \\ 2 & 3 & 1 & 6 & 2 & 2 \\ 7 & 4 & 5 & 9 & 1 & 0 \\ 4 & 1 & 9 & 0 & 9 & 1 \\ \hline \end{array} (54)\ +} \qquad \underset{\textbf{B}}{\overset{(64)\ +}{\underset{(66)\ -}{\begin{array}{|cccccc|} \hline 1 & 8 & 4 & 6 & 4 & 0 \\ 3 & 1 & 2 & 4 & 0 & 1 \\ 6 & 7 & 8 & 3 & 3 & 3 \\ \hline 2 & 3 & 1 & 6 & 2 & 2 \\ 7 & 4 & 5 & 9 & 1 & 0 \\ 4 & 1 & 9 & 0 & 9 & 1 \\ \hline \end{array}}}}$$

Fig. 22. Comparison of matrix sectors before balancing

For an illustrative example, the sectors of the matrix, which are separated by a boundary, were identified. Numbers in parentheses are the sum of all numbers in a sector. Accordingly, the plus and minus signs mean the increase and decrease of sectors, namely all numbers of a continuous sector.

$$\underset{\textbf{A}}{\begin{array}{|ccc|ccc|} \hline 0 & 7 & 3 & 7 & 5 & 1 \\ 2 & 0 & 1 & 5 & 1 & 2 \\ 5 & 6 & 7 & 4 & 4 & 4 \\ 1 & 2 & 0 & 7 & 3 & 3 \\ 6 & 3 & 4 & 0 & 2 & 1 \\ 3 & 0 & 8 & 1 & 0 & 2 \\ \hline \end{array}} \qquad \underset{\textbf{B}}{\begin{array}{|cccccc|} \hline 2 & 9 & 5 & 7 & 3 & 1 \\ 4 & 2 & 3 & 5 & 1 & 2 \\ 7 & 8 & 9 & 4 & 4 & 4 \\ \hline 1 & 2 & 0 & 5 & 1 & 1 \\ 6 & 3 & 4 & 8 & 0 & 9 \\ 3 & 0 & 8 & 9 & 8 & 0 \\ \hline \end{array}}$$

Fig. 23. The matrix is balanced horizontally and vertically

The figure shows that the variants of the matrix after horizontal and vertical balancing differ from each other by only half. This is not a flaw or error of the algorithm. Such cases occur in algorithms of this level when they are used for paired uniform (symmetric matrices).

Even if you replace the virtual boundary of the matrix with a defined one, the result will be the same, the matrices after both types of balancing will differ by exactly half, except for the invariant boundary in the form of a row or column of the matrix. Therefore, there is actually no point in conducting an additional experiment, especially if the main result of the demonstration is known in advance.

Several more combined algorithms are based on this algorithm, but they belong to algorithms of another (higher level), so they will not be described in this article.

These three levels of balancing algorithms that were shown in the article are just the tip of the iceberg, in fact there are many more. In this paper, the algorithms of the initial level are described, and in the following publications the algorithms of other (higher) levels will be shown and described.

It should also be noted that each algorithm of each level is unique, useful and efficient, despite its simplicity.

4. Conclusions

This paper presents three levels of algorithms for balancing numerical matrices. These levels belong to the class of low (initial or basic) levels, because all higher levels in one way or another originate (basis) from them or derivative algorithms.

The uniqueness of this development is that with the right combination of these algorithms, we get a new and effective method of encryption (other areas of science are not excluded). Some algorithms and their combinations have proven themselves at a high level in experiments with data encryption. This cipher is almost impossible to break, because it does not match the generally accepted rules or algorithms. It is known that ideal methods, algorithms or systems do not exist in nature. However, this method is not amenable to statistical or frequency analysis and any patterns in it can not be detected.

After describing all the classes, each of which consists of several levels, which in turn consist of sets of algorithms and their variations, the author's further plans include the development of this field of science. Publications are planned, which will show and describe the application of these algorithms in practice (so far for encryption), search for combinations of developed algorithms, description of methodological support (rules, comments, definitions of abbreviations), demonstration of inverse (inverse) algorithms for decoding, presentation of mathematical formulas and mathematical models of the method, the use of special markers for the formation of encryption keys.

References

[1] Agarwal S.: Symmetric Key Encryption using Iterated Fractal Functions. International Journal of Computer Network and Information Security 9(4)/2019, 1–9.

[2] Anisimov A. V., Kulyabko P. P.: Information systems and databases: A textbook for students of the faculty computer science and cybernetics. Kyiv 2017.

[3] Bogdanov A., Khovratovich D., Rechberger D.: Biclique Cryptanalysis of the Full AES. Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science 7073. Springer, Berlin 2011.

[4] Borecki M.: Risk level analysis in the selected (initial) stage of the project life cycle. Management and production engineering review 11(4)/2020, 104–112.

[5] Borecki M., Ciuba M., Kharchenko Y., Khanas Y.: Main aspects influencing the evaluation of atmospheric overvoltages in high-voltage networks. Bull. Pol. Ac.: Tech 69(1)/ 2021, 1–8.

[6] Buryachok V. L.: The choice of a rational method of generating passwords among many existing. Information security 25(1)/2019, 59–64.

[7] Buryachok V. L.: Generate a password for wireless networks using variable rule of complication. Information protection 21(1)/2019, 52–59.

[8] Hughes J., Cybenko G.: Quantitative Metrics and Risk Assessment: The Three Tenets Model of Cybersecurity. Technology Innovation Management Review 2013, 15–24.

[9] Isa M. A. M., Hashim H., Ab Manan J. L., Adnan S. F. S., Mhmod R.: RF simulator for cryptographic protocol. IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2014, 518–523.

[10] Josefsson S., Leonard S.: Textual Encodings of PKIX, PKCS, and CMS Structures. Internet Engineering Task Force April 2015.

[11] Khanas, Y., Borecki, M.: Research on the use of algorithms for matrix transformations for encrypting text information. Security and Privacy 3(6)/2020, 1–13.

[12] Khanas Y., Ivanciv R., Litvinko S.: The algorithm for minimizing matrices in the given direction of reduction and the rules for their restoration. Visnyk of the National University "Lviv Polytechnic" 882/2017, 12–17.

[13] Khanas Y., Ivanciv R.: Application Mirroring of Matrices to Prevent Excessive Reduction. Perspective technologies and design methods of MEMST (MEMSTECH 2016), Lviv-Polyana 2016, 143–145.

[14] Krasilenko V. G.: Multifunctional parametric matrix-algebraic models (MAM) of cryptographic transformations (CP) with modular operations and their modeling. 72 NPK – conference materials, Odessa 2017, 123–128.

[15] Krasilenko V. G.: Improvement and modeling of electronic digital signatures of matrix type for textographic documents. Proceedings of the VI International Scientific and Practical Conference "Information Control Systems and Technologies" (IUST-Odessa-2017), Odessa 2017.

[16] Krasylenko V. G., Nikitovich D. V., Yatskovskaya R. O., Yatskovsky V. I.: Simulation of advanced multi-step 2D RSA algorithms for cryptographic transformations and blind electronic digital signature. Information processing systems 1(156)/2019, 92–100.

[17] Leurent G., Peyrin T.: SHA-1 is a Shambles. First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust. Real World Crypto 2020.

[18] Lobur M. V., Ivantsiv R. D., Kolesnyk K. K., Khanas Y. Y.: Development of an algorithm for reducing matrices depending on their size and content. Scientific and Technical Journal "Instrumentation Technology" 2/2016, 29–31.

[19] Nazarkevich M. A., Dronyuk I. M., Troyan O. A., Tomashchuk T. Yu.: Development of a method of document protection by latent elements based on fractals. Information protection 17(1)/2015, 21–26.

[20] Nikonov V. G., Zobov A. I.: On the possibility of using fractal models in the construction of information security systems. Computational nanotechnology 1/2017, 39–49.

[21] Ortiz S. M., Parra O., Miguel J., Espitia R.: Encryption through the use of fractals. International Journal of Mathematical Analysis 11(21)/2017, 1029–1040.

[22] Wenliang Du: Computer Security: A Hands-on Approach. CreateSpace Independent Publishing Platform, 2017.

M.Sc. Yuriy Khanas
e-mail: yuriy.y.khanas@lpnu.ua

Yuriy Khanas is an assistant at the Department of Computer-Aided Design, Lviv Polytechnic National University, Lviv, Ukraine. His main research interests include algorithms and data compression and encryption systems based on matrix transformation algebra, information protection systems and algorithms.

http://orcid.org/0000-0001-6496-5782

Ph.D. Michał Borecki
e-mail: michal.borecki@ee.pw.edu.pl

Michał Borecki received the Ph.D. degree in electrical engineering from Warsaw University of Technology, Poland, in 2017. He is an assistant professor in Department of High Voltage and Electromagnetic Compatibility.
Author of many articles in the field of lightning protection, power network and risk managemenet.

http://orcid.org/0000-0001-8907-6906

otrzymano/received: 04.03.2021 *przyjęto do druku/accepted: 15.03.2021*