UDC 519.718
DOI: 10.15587/1729-4061.2021.225325

DEVELOPING AN ALGORITHM TO MINIMIZE BOOLEAN FUNCTIONS FOR THE VISUAL-MATRIX FORM OF THE ANALYTICAL METHOD

M. Solomko
PhD, Associate Professor
Department of Computer Engineering
National University of Water and Environmental Engineering
Soborna str., 11, Rivne, Ukraine, 33028
E-mail: doctrinas@ukr.net

This research has established the possibility of improving the effectiveness of the visual-matrix form of the analytical Boolean function minimization method by identifying reserves in a more complex algorithm for the operations of logical absorption and super-gluing the variables in terms of logical functions.

An improvement in the efficiency of the Boolean function minimization procedure was also established, due to selecting, according to the predefined criteria, the optimal stack of logical operations for the first and second binary matrices of Boolean functions. When combining a sequence of logical operations using different techniques for gluing variables such as simple gluing and super-gluing, there are a small number of cases when function minimization is more effective if an operation of simply gluing the variables is first applied to the first matrix. Thus, a short analysis is required for the primary application of operations in the first binary matrix. That ensures the proper minimization efficiency regarding the earlier unaccounted-for variants for simplifying the Boolean functions by the visual-matrix form of the analytical method. For a series of cases, the choice of the optimal stack is also necessary for the second binary matrix.

The experimental study has confirmed that the visual-matrix form of the analytical method, whose special feature is the use of 2-(n, b)-design and 2-(n, x/b)-design systems in the first matrix, improves the process efficiency, as well as the reliability of the result of Boolean function minimization. This simplifies the procedure of searching for a minimal function. Compared to analogs, that makes it possible to improve the productivity of the Boolean function minimization process by 100–200 %.

There is reason to assert the possibility of improving the efficiency of the Boolean function minimization process by the visual-matrix form of the analytical method, through the use of more complex logical operations of absorbing and super-gluing the variables. Also, by optimally combining the sequence of logical operations of super-gluing the variables and simply gluing the variables, based on the selection, according to the established criteria, of the stack of logical operations in the first binary matrix of the assigned function

Keywords: Boolean function minimization, visual-matrix form of analytical method, binary matrix

Received date 29.12.2020
Accepted date 02.02.2021
Published date 26.02.2021

1. Introduction

The analytical method, which is based on equivalent transformations by means of laws and equalities of Boolean algebra, is effective in simplifying relatively simple Boolean functions. The essence is to move from PDNF (PKNF) to DNF (KNF) at a minimum number of terms. The number of literals in each term should also be minimal. The disadvantage of the analytical method is the uncertainty in the sequence of logical operations in the simplification of functions, and, consequently, the lack of an algorithm for minimizing Boolean functions. In turn, the absence of an algorithm does not always warrant that the resulting expression of the Boolean function would be minimal, implying the impossibility of further simplification.

Two forms of information, which are determined by reflective and continual thinking, are presented in paper [1].

In the first case, a person receives information in words, thinks in words, and, sometimes, converts them into images. This method of information transmission (verbal) has a small information capacity, requires the active participation of brain structures to decipher, process, and supplement the information received.

In continual consciousness, thinking occurs not in words but in images. Figurative thinking is characterized by a large inflow of information per unit of time, incomparable with verbal.

Work [2] states that «information is transferred not only by books strewn with letters or by human speech but also sunlight, folds of a mountain range, the noise of a waterfall, and the rustle of grass».

The visual-matrix Boolean function minimization methods were considered in studies [3–6] and others.

The verbal form of information is the most common variant of representing a description of the subject area. Any algebraic expression is primarily a text formed according to certain rules. The algebraic technique of Boolean function minimization is a verbal procedure. Illustrations are turned to when there is an attempt to explain what is difficult to express by text.

An illustrative (figurative) description is visual, which makes it possible to simultaneously represent a system of relations among the individual variables of a task. A characteristic feature of an image is its semantic capacity, the ability to transmit a large amount of information with a small number of characters and, as a result, ensuring the implication of a significant part of this information [7].

Thus, the figurative form of information in the form of combinatorial objects should provide for a better chance to determine the algorithm of the analytical method of Boolean function minimization. Combinatorial objects, in this case, are the two-dimensional binary matrices and the presence in the structure of truth table of complete repeated 2-(n, b)-design, or incomplete 2-(n, x/b)-design systems, as well as the combinatorial images themselves [8–10]. As a result, the verbal procedures of algebraic transformations are replaced by equivalent figurative transformations [8].

Example 1: It is required to minimize the function $f(x_1, x_2, x_3, x_4)$ (1) [11] using the Blake-Poretsky method.

$$f(x_1,x_2,x_3,x_4)=\bar{x}_1x_2+x_2\bar{x}_3x_4+\bar{x}_1x_3x_4+x_2x_3x_4. \qquad (1)$$

The blake-Poretsky method is based on the use of a logical operation of the generalized gluing of variables:

$$A\cdot x+B\cdot\bar{x}=A\cdot x+B\cdot\bar{x}+A\cdot B,$$

which makes it possible to find a minimum function according to the arbitrary DNF of its representation. The second and fourth variables of the function $f(x_1, x_2, x_3, x_4)$ (1) allow the generalized gluing for variable x_3.

$$x_2\bar{x}_3x_4+x_2x_3x_4=x_2\bar{x}_3x_4+x_2x_3x_4+x_2x_4.$$

Obviously, no other variables of the function $f(x_1, x_2, x_3, x_4)$ (1) permit the generalized gluing for other variables.

Upon completing the last generalized gluing, one obtains:

$$f(x_1,x_2,x_3,x_4)=\bar{x}_1x_2+x_2\bar{x}_3x_4+\bar{x}_1x_3x_4+x_2x_3x_4+x_2x_4.$$

Following all absorptions, one obtains a minimum function:

$$f(x_1,x_2,x_3,x_4)=\bar{x}_1x_2+\bar{x}_1x_3x_4+x_2x_4. \qquad (2)$$

Further use of the generalized operations of gluing the variables and absorptions does not yield results.

Minimizing the function $f(x_1, x_2, x_3, x_4)$ (1) [11] with equivalent figurative transformations takes the following form:

$$f(x_1,x_2,x_3,x_4)=\begin{array}{|cccc|}\hline x_1 & x_2 & x_3 & x_4\\ \hline 0 & 1 & & \\ & 1 & 0 & 1\\ 0 & & 1 & 1\\ & 1 & 1 & 1\\ \hline\end{array}=$$
$$=\begin{vmatrix}0 & 1 & & \\ 0 & & 1 & 1\\ & 1 & & 1\end{vmatrix}=\bar{x}_1x_2+\bar{x}_1x_3x_4+x_2x_4.$$

The minimum function, derived from the operation of simple gluing of variables, is:

$$f(x_1,x_2,x_3,x_4)=\bar{x}_1x_2+\bar{x}_1x_3x_4+x_2x_4. \qquad (3)$$

It is also possible to simplify function (1) by means of the operation of the generalized gluing of variables:

$$f(x_1,x_2,x_3,x_4)=\begin{array}{|cccc|}\hline x_1 & x_2 & x_3 & x_4\\ \hline 0 & 1 & & \\ & 1 & 0 & 1\\ 0 & & 1 & 1\\ & 1 & 1 & 1\\ \hline\end{array}=\begin{vmatrix}0 & 1 & & \\ & 1 & 0 & 1\\ 0 & & 1 & 1\\ & 1 & 1 & 1\\ & 1 & & 1\end{vmatrix}=$$
$$=\begin{vmatrix}0 & 1 & & \\ 0 & & 1 & 1\\ & 1 & & 1\end{vmatrix}=\bar{x}_1x_2+\bar{x}_1x_3x_4+x_2x_4.$$

The alternation of zeros and unities in the third column of the first matrix is a hyper parament (more than a prerequisite) for the operation of generalized gluing, for variable x_3. In the second matrix, the operation of the absorption of variables was carried out. The results of the minimization (2) and (3) coincide but the method of figurative transformations is a much simpler procedure for simplifying a function.

The visual-matrix technique of Boolean function minimization is, to some extent, a complete and independent method based on the use of some properties of the visual perception of information [3–6]. The most compact form of information representation, in this case, is a two-dimensional matrix. The order of the mutual arrangement of matrix elements, the same under an algebraic approach, plays an essential role in the visual perception of two-dimensional data. The potential possibilities of minimizing Boolean functions represented by the matrix are provided by the properties of frequency and symmetry of the image [3, 5]. Although visual-matrix methods became known in the late 1940s-early 1950s, the visual method, based on binary combinatoric systems with the repeated 2-(n, b)-design, 2-(n, x/b)-design, representing something separate from the frequency and symmetry of matrix image, the product of a specific assumption, was launched and has been developing since 2017 [8].

Paper [12] considers the consistent alternation of logical operations of super-gluing the variables (if such an operation is possible) and simple gluing of variables in the first matrix of the Boolean function, which ensures the efficiency of minimization and is the basis for the algorithmization of the analytical method. The authors demonstrated an example of minimizing the 4-bit Boolean function when the consistent use of these logical operations is not always optimal in terms of the procedure effectiveness.

The evolution of the analytical method of simplification of logical functions and its algorithmization is the result of continuous optimization, so it has remained relevant to undertake research aimed, in particular, at making an update to the minimization algorithm by the analytical method, considered in work [12]. That would render proper efficiency to the previously unaccounted-for variants for simplifying Boolean functions by the analytical method, in particular in the class of PDNF and PKNF representations, as well as make it possible to optimize the cost of technology of Boolean function minimization by the analytical method.

2. Literature review and problem statement

A new method of minimizing logical functions at a relatively small number of variables is proposed in work [13].

The method implies the inclusion of the ordering of the optimal minterms of the logical function. The selected minterms and a minimization procedure ensure optimal coverage. This type of optimal arrangement helps the minimization algorithm quickly cover the necessary minterms, which reduces the complexity of the assigned functions in the end. The method is also designed and used to minimize complex Boolean functions. Comparative analysis showed that the proposed technique of including the procedure for organizing optimal minterms makes it possible to obtain more accurate simplification results; it uses less time and memory compared to the ESPRESSO software. The practice of organizing optimal minterms to minimize Boolean functions is given.

Approximate synthesis is a recent trend of logical synthesis when some results of the logical specification change within the permissible error of the assigned application in order to reduce complexity and accelerate the implementation of the final digital component; it was considered in paper [14]. The task of synthesis is solved by using the permissible flexibility of technology to maximize the regularity of the assigned Boolean functions. In particular, the authors consider two types of regularity: symmetry and D-contractility. Two algorithms are implemented in order to find, respectively, symmetrical and D-contracted approximation of the assigned function, within the permissible error, if possible. When pointing to symmetry, the technology characterizes and computes the nearest symmetrical approximation. The authors presented a polynomial heuristic algorithm to calculate the D-approximation of the incompletely assigned Boolean function according to the bitmap error metric. Experimental results on the classical and new standards confirm the effectiveness of the proposed approaches.

The issue of covering the logical structure and minimizing PLA is considered in paper [15]. The issue of minimal coverage is resolved by using the so-called implicit enumeration method. The specified method in paper [15] is a modification of the Quine-McClusky method, adapted for computer processing. The method has extensions that use some new properties of minimal coverage algorithms to speed up the procedure. To solve large-scale problems, a heuristic algorithm is presented. Its application to minimize programmable logical arrays (PLA) is shown as an example. The authors discuss the computational experience presented to confirm the improvement of the tasks of minimal coverage by the implicit enumeration method.

The software implementation of the Quine-McClusky method for minimizing Boolean expressions is considered in paper [16]. The Quine-McCluskey (QM) method is one of the most powerful techniques for simplifying Boolean expressions. Compared to other methods, the QM method can handle logical functions with a larger number of variables. In addition, the QM method is easier to implement in computer programs, which makes it an effective technique. For the QM algorithm of Boolean function minimization, the C-language code is presented.

A new fast method for minimizing Boolean functions is reported in work [17]. The method employs neural networks (FNN) implemented in the frequency domain. The network is entrusted with the execution of cross-correlation in the frequency domain. It was proven by calculations and confirmed in practice that the number of computation stages required for FNN networks is less than is necessary for standard neural networks (CNN). Modeling results using MATLAB confirm theoretical calculations.

Advanced QMC algorithm (*e*QMC), which improves the productivity of the Boolean function minimization process by the Quine-McClusky method, is described in work [18]. The authors demonstrated the increased speed and performance of the computer's memory operation by simulating the process of Boolean function minimization.

Over the past few decades, much effort has been made in the field of QCA methodology to develop effective algorithms for minimizing Boolean functions in order to obtain an accurate, and most importantly, a complete list of minimal simple implicants. As the complexity of the method increases exponentially with each new state, the required computer memory is past the current computer resources and the polynomial time required to solve this problem. Article [19] presents a new alternative to existing non-polynomial attempts that fully solves the memory issue. Previous tests show that the problem is resolved hundreds of times faster compared to the *e*QMC method. Although speed is not a big issue at the moment (the *e*QMC algorithm is fast enough for simple data), it may prove important when further developing to all possible time orders or searching for configurations in the panel data over time, combined with/or automatic detection of complex facts of the procedure, etc.

The considered literary sources [13–19] mainly report complex algorithms for minimizing Boolean functions based on the Quine-McClusky method, its modifications, and cubic technique. Compensation for the complexity of the search for the optimal function for such algorithms may be an approximate synthesis – the tendency of logical synthesis when some results of the logical specification change within the permissible optimality of the digital circuit being designed. A regular technological point for such algorithms is the comparison of the result with the result of minimization by the heuristic ESPRESSO algorithm. Although ESPRESSO does not warrant that the result of minimization would be a global minimum, in practice it is very close, and, at the same time, the solution is always devoid of redundancy [20].

The visual-matrix form of the analytical method, based on the binary combinatorial systems with the repeated 2-(n, b)-design, 2-(n, x/b)-design, as somewhat separate from the complex Quine-McClusky algorithms and cubic technique, does not imply an approximate minimization result and does not exclude the manual method of Boolean function minimization. Algorithms for automating the procedure of simplification of logical functions by the analytical method are presented in [12].

Thus, the complex Quine-McClusky algorithms, their modifications, cubic technique, software tools developed for them, covering the general procedure for minimizing logical functions [13–19], and the visual-matrix form of the analytical method, follow different approaches (principles of minimization). Therefore, they imply different prospects regarding the possibility of algorithmic minimization of logical functions.

Such components of the technology as «approximate synthesis», «heuristic algorithm», «adaptation to computer processing» are technological tools for current practice, which is changing. This is the reason to believe that the software and technological base, represented by the Quine-McClusky algorithms, their modifications, and cubic technique [13–19], is insufficient (in a state of change) for theoretical research on the optimal Boolean function minimization. That necessitates research into the visual-matrix form of the analytical method of minimizing logical functions. In particular, the peculiarities of relatively complex algorithms for logical operations of absorption and super-gluing the variables, a stack of logical operations for the first binary matrix of the assigned function, techniques for the simplification of the general form by using Boolean functions.

In applied terms, this approach would expand the capabilities of technology in designing digital components. Although the Quine-McClusky algorithm is well suited for software implementation, the result is still inefficient in terms of computing time and memory usage. The addition of a variable to a function roughly doubles both, since the size of the truth table increases exponentially with the growth of the number of variables [20]. With an increase in the bit size of Boolean functions, the data array, not suitable for simplification, increases, while the 2-(n, b)-design, 2-(n, x/b)-design systems in the binary configuration of the function are placed less frequently (example 12). The greater the bit size of logical functions, the lower the efficiency of the Quine-McClusky algorithm. In turn, the visual-matrix procedure primarily finds the 2-(n, b)-design, 2-(n, x/b)-design systems, followed by the minimization. Therefore, with an increase in the bit size of logical functions, the effectiveness of the visual-matrix procedure does not decrease.

3. The aim and objectives of the study

The purpose of this study is to establish an updated procedure for alternating the protocols of equivalent transformations for the first and second matrices of the assigned logical function, compared to what was considered in [12]. This would make it possible to define the new standard of the procedure, which could ensure proper efficiency for the previously unaccounted-for variants for simplifying Boolean functions by analytical method, specifically in the class of PDNF and PKNF representations.

To accomplish the aim, the following tasks have been set:

– to establish patterns in using the logical operations of absorption and super-gluing the variables to minimize Boolean functions by the analytical method;

– to determine the stack of logical operations under the analytical method for minimizing the assigned logical function within the first and second matrices;

– to analyze methods of simplifying the Boolean functions of the general form;

– to demonstrate examples of Boolean function minimization in order to assess the effectiveness of the visual-matrix form of the analytical method at minimizing logical functions.

4. Features in using the logical operations of absorption and super-gluing the variables

Logical operations with variables on the binary and algebraic function structures, to a certain extent, would be highlighted in color. That could provide better didactics of the method.

For the analytical method of Boolean function minimization, the absorption of variables with duplication of elementary conjunctions may take the following form:

$$\begin{vmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{vmatrix}\begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & & \\ & 0 & 0 & & 1 \end{vmatrix} = \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & \\ 1 & & 0 & 1 & \\ & 0 & 0 & & 1 \end{vmatrix} = \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & \\ & 0 & 0 & & 1 \end{vmatrix} =$$

$$= \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & 0 \\ 1 & & 0 & 0 & 1 \\ & 0 & 0 & & 1 \end{vmatrix} = \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & & 0 & 0 & 1 \\ & 0 & 0 & & 1 \end{vmatrix} = \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ & 0 & 0 & & 1 \end{vmatrix} =$$

$$= \begin{vmatrix} 1 & & & 1 & \\ 1 & & & & 0 \\ 1 & 1 & & & \\ & 0 & 0 & & 1 \end{vmatrix}.$$

$$x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3 + \overline{x}_2\,\overline{x}_3x_5 =$$
$$= x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\,\overline{x}_4 + x_1\overline{x}_3x_4 + \overline{x}_2\,\overline{x}_3x_5 =$$
$$= x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\,\overline{x}_4 + \overline{x}_2\,\overline{x}_3x_5 =$$
$$= x_1x_4 + x_1\overline{x}_5 + x_1x_2 + x_1\overline{x}_3\,\overline{x}_4\,\overline{x}_5 +$$
$$+ x_1\overline{x}_3\,\overline{x}_4x_5 + \overline{x}_2\,\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 +$$
$$+ x_1\overline{x}_3\,\overline{x}_4x_5 + \overline{x}_2\,\overline{x}_3x_5 = x_1x_4 + x_1\overline{x}_5 + x_1x_2 +$$
$$+ x_1\overline{x}_2\,\overline{x}_3\,\overline{x}_4x_5 + x_1x_2\overline{x}_3\,\overline{x}_4x_5 + \overline{x}_2\,\overline{x}_3x_5 =$$
$$= x_1x_4 + x_1\overline{x}_5 + x_1x_2 + \overline{x}_2\,\overline{x}_3x_5.$$

Duplication of the constituents followed by the logical operation of the simple gluing of variables may take, for example, the following form:

$$\begin{vmatrix} 0 & & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & & 1 \end{vmatrix} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & & 1 \end{vmatrix} = \begin{vmatrix} 0 & 1 & \\ & 0 & 0 \\ 1 & & 1 \end{vmatrix}. \tag{4}$$

It is possible to derive another minimal expression when duplicating the constituent for the lower row of the first binary matrix (4).

Duplication of the constituents followed by the operation of super-gluing the variables may take, for example, the following form:

$$\begin{vmatrix} 1 & 1 & 0 & & 1 & 0 \\ 1 & 1 & 1 & & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 & & 1 & 0 \\ 1 & 1 & 1 & & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{vmatrix} =$$

$$= \begin{vmatrix} 1 & 1 & 0 & & 1 & 0 \\ 1 & 1 & 1 & & 0 & 0 \\ 1 & 1 & 0 & 1 & & \\ 1 & 1 & 1 & 0 & & \end{vmatrix}. \tag{5}$$

$$
\begin{aligned}
& x_1x_2\overline{x_3}x_5\overline{x_6}+x_1x_2x_3\overline{x_5}\,\overline{x_6}+x_1x_2\overline{x_3}x_4\overline{x_5}\,\overline{x_6}+\\
&+x_1x_2\overline{x_3}x_4\overline{x_5}x_6+x_1x_2\overline{x_3}x_4x_5x_6+\\
&+x_1x_2x_3\overline{x_4}\,\overline{x_5}x_6+x_1x_2x_3\overline{x_4}x_5\overline{x_6}+\\
&+x_1x_2x_3\overline{x_4}x_5x_6=x_1x_2\overline{x_3}x_5\overline{x_6}+x_1x_2x_3\overline{x_5}\,\overline{x_6}+\\
&+x_1x_2\overline{x_3}x_4\overline{x_5}\,\overline{x_6}+x_1x_2\overline{x_3}x_4\overline{x_5}x_6+x_1x_2\overline{x_3}x_4x_5x_6+\\
&+x_1x_2\overline{x_3}x_4x_5\overline{x_6}+x_1x_2x_3\overline{x_4}\,\overline{x_5}x_6+x_1x_2x_3\overline{x_4}x_5\overline{x_6}+\\
&+x_1x_2x_3\overline{x_4}x_5x_6+x_1x_2x_3\overline{x_4}\,\overline{x_5}\,\overline{x_6}=x_1x_2\overline{x_3}x_5\overline{x_6}+\\
&+x_1x_2x_3\overline{x_5}\,\overline{x_6}+x_1x_2\overline{x_3}x_4\left(\overline{x_5}\,\overline{x_6}+\overline{x_5}x_6+x_5x_6+x_5\overline{x_6}\right)+\\
&+x_1x_2x_3\overline{x_4}\left(\overline{x_5}x_6+x_5\overline{x_6}+x_5x_6+\overline{x_5}\,\overline{x_6}\right)=\\
&=x_1x_2\overline{x_3}x_5\overline{x_6}+x_1x_2x_3\overline{x_5}\,\overline{x_6}+x_1x_2\overline{x_3}x_4+\\
&+x_1x_2x_3\overline{x_4}. \qquad (6)
\end{aligned}
$$

The results of the matrix (5) and algebraic (6) techniques of minimizing a logical expression coincide.

The reported relatively more complex algorithms for the application of logical operations of absorption and super-gluing of variables expand the variants for their use, which improves the efficiency of the Boolean function minimization procedure by the analytical method.

5. A stack of logical operations in the first and second binary matrices of the analytical method

The effectiveness of minimizing Boolean functions by the analytical method depends on combining a sequence of logical operations involving different techniques for gluing the variables – simple gluing and super-gluing in the first matrix, and, in some cases, in the second binary matrix as well. To evaluate the result of transformations in the first matrix, based on the selected stack (list) of logical operations, one can set the criterion for the effectiveness of the conversion of logical expressions. One can evaluate the result of the transformation of the first matrix by the number of remaining terms. With an equal number of terms, the criterion can be the number of literals. With equality of the number of terms and literals, the criterion may be the number of inverted variables.

Example 2: It is required to select the optimal stack of logical operations in the first matrix of the 4-variable Boolean function from eight sets represented in PDNF.

The first stack of logical operations: simple gluing of variables.

$$
\left|\begin{array}{cccc}
0&0&1&1\\
0&1&0&1\\
1&0&0&0\\
1&0&0&1\\
1&0&1&1\\
1&1&0&1\\
1&1&1&0\\
1&1&1&1
\end{array}\right|
=
\left|\begin{array}{cccc}
 &0&1&1\\
 &1&0&1\\
1&0&0& \\
1&1&1&
\end{array}\right|
$$

The second stack of logical operations: super-gluing the variables.

$$
\left|\begin{array}{cccc}
0&0&1&1\\
0&1&0&1\\
1&0&0&0\\
1&0&0&1\\
1&0&1&1\\
1&1&0&1\\
1&1&1&0\\
1&1&1&1
\end{array}\right|
=
\left|\begin{array}{cccc}
0&0&1&1\\
0&1&0&1\\
1&0&0&0\\
1& & &1\\
1&1&1&0
\end{array}\right|
=
\left|\begin{array}{cccc}
 &0&1&1\\
 &1&0&1\\
1&0&0& \\
1& & &1\\
1&1&1&
\end{array}\right|
=
$$

$$
=
\left|\begin{array}{cccc}
 &0&1&1\\
 &1&0&1\\
1&0&0& \\
1&0&0&1\\
1&1&0&1\\
1&0&1&1\\
1&1&1&1\\
1&1&1&
\end{array}\right|
=
\left|\begin{array}{cccc}
 &0&1&1\\
 &1&0&1\\
1&0&0& \\
1&1&1&
\end{array}\right|
$$

Two stacks of logical operations yield the same result of minimizing the Boolean function. However, the conversion of the first stack results in four terms (rewritten in the second matrix), and the conversion of the second stack results in five terms. Therefore, the optimal conversion, according to the criterion, is the conversion based on the first stack of logical operations.

The considered binary configuration of the Boolean function is a rare case when logical operations of simple gluing of variables are optimal to minimize the function. In most cases, optimal transformations in the first matrix are carried out through the operation of super-gluing the variables [12].

Example 3: It is required to minimize the logical function $F(x_1, x_2, x_3, x_4)$ by the analytical method, which is given in the canonical form [13]:

$$F=\Sigma\,(4, 7, 9, 10, 12, 13, 14, 15). \qquad (7)$$

Note: the values in Σ are the minterms for the rows of the truth table when the function $F(x_1, x_2, x_3, x_4)$ returns «1» at the output.

After analyzing the binary configuration of the function $F(x_1, x_2, x_3, x_4)$ (7) truth table (7), it is concluded that the optimal stack of logical operations for equivalent transformations in the first matrix is operations of the simple gluing of variables. Then the simplification of the function $F(x_1, x_2, x_3, x_4)$ (7) [13] by the analytic method takes the following optimal form:

$$
F\left(x_1,x_2,x_3,x_4\right)=
$$

$$
=
\left|\begin{array}{c|cccc}
4&0&1&0&0\\
7&0&1&1&1\\
9&1&0&0&1\\
10&1&0&1&0\\
12&1&1&0&0\\
13&1&1&0&1\\
14&1&1&1&0\\
15&1&1&1&1
\end{array}\right|
=
\left|\begin{array}{cccc}
 &1&0&0\\
 &1&1&1\\
1& &0&1\\
1& &1&0
\end{array}\right|
=
$$

$$
=x_2\overline{x_3}\,\overline{x_4}+x_2x_3x_4+x_1\overline{x_3}x_4+x_1x_3\overline{x_4}. \qquad (8)
$$

The result of minimizing (8) the function $F(x_1, x_2, x_3, x_4)$ (7) coincides with the result of minimizing when using the method of the optimal ordering of minterms [13] but the minimization procedure involving the analytical method is simpler.

Example 4: It is required to choose the optimal stack of logical operations to simplify the Boolean function $F(x_1, x_2, x_3)$ [21], represented in the algebraic form.

$$f(x_1,x_2,x_3)=\overline{x}_1\,\overline{x}_2\,\overline{x}_3+x_1\overline{x}_2\,\overline{x}_3+\overline{x}_1x_2\overline{x}_3+$$
$$+x_1\overline{x}_2x_3+\overline{x}_1x_2x_3+x_1x_2x_3. \tag{9}$$

Solution:

To simplify $f(x_1, x_2, x_3)$ (9), apply the duplication of the constituents followed by the operation of the simple gluing of variables (p. 4):

$$f(x_1,x_2,x_3)=\begin{vmatrix}0&0&0\\1&0&0\\0&1&0\\1&0&1\\0&1&1\\1&1&1\end{vmatrix}=\begin{vmatrix}0&0&0\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&1\end{vmatrix}=\begin{vmatrix}0&&0\\0&1&1\\1&0&0\\1&&1\end{vmatrix}=$$
$$=\begin{vmatrix}0&0&0\\0&1&0\\0&1&1\\1&0&0\\1&&1\end{vmatrix}=\begin{vmatrix}0&1&\\&0&0\\1&&1\end{vmatrix}. \tag{10}$$

Sets of variables in the first matrix (10) are recorded in lexicographic order. Such arrangement should always be done when using the visual-matrix form of the analytical method. The result of ordering is rewritten to the second matrix (10). In the second matrix (10), implement two operations of the simple gluing of variables, after which the result of gluing the variables is rewritten to the third matrix (10). Next, apply the duplication of the constituents (highlighted in red). In the fourth matrix, perform two operations of the simple gluing of variables. The result is the following minimum function:

$$f(x_1,x_2,x_3)=\overline{x}_1x_2+\overline{x}_2\,\overline{x}_3+x_1x_3. \tag{11}$$

The result of minimizing (11) the function $F(x_1, x_2, x_3)$ (9) coincides with the result of minimization when using a distance matrix [21]. For the example in question, the procedure for minimizing via the analytical method is simpler.

Example 5: It is required to choose the optimal stack of logical operations for equivalent transformations in the first and second binary matrices of the Boolean function $F(x_1, x_2, x_3, x_4, x_5)$ (Fig. 1) [12], represented in PDNF.

The effectiveness of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$ (Fig. 1) is based on the primary application of the logical operation of super-gluing the variables in the first matrix. Super-gluing the variables is possible for blocks: 1, 3, 5, 7 (highlighted in green), 4, 12, 20, 28 (highlighted in red), 14, 15, 30, 31 (highlighted in blue). These blocks are a complete binary combinatory system with the repeated 2-(2, 4)-design [9]. The logical operation of the simple gluing of variables is not used in the first matrix. The result of minimization in the first matrix by means of the logical operations of super-gluing the variables is rewritten to the second matrix.

Minimization of the function in the second matrix is also carried out by means of super-gluing the variables, however, to form the 2-(2, 4)-design systems, duplication of the corresponding constituents (p. 4) must be applied. The result of duplicating the constituents is rewritten to the third matrix. Further, the minimization of the function is carried out by simply gluing the variables, duplicating the elementary conjunctions, and variable absorption operations.

The result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$ (Fig. 1) coincides with the result of minimizing in [12]. In each case, the minimization of the function $F(x_1, x_2, x_3, x_4, x_5)$ (Fig. 1) is carried out efficiently, although the minimization procedures use different protocols.

When combining a sequence of logical operations using different techniques of gluing the variables such as simple gluing and super-gluing, there are a small number of cases when the minimization of the function is more effective if the operation of simply gluing the variables is first applied in the first matrix [12].

$$\begin{vmatrix}1&0&0&0&0&1\\2&0&0&0&1&0\\3&0&0&0&1&1\\4&0&0&1&0&0\\5&0&0&1&0&1\\7&0&0&1&1&1\\9&0&1&0&0&1\\11&0&1&0&1&1\\12&0&1&1&0&0\\13&0&1&1&0&1\\14&0&1&1&1&0\\15&0&1&1&1&1\\16&1&0&0&0&0\\17&1&0&0&0&1\\18&1&0&0&1&0\\20&1&0&1&0&0\\22&1&0&1&1&0\\26&1&1&0&1&0\\28&1&1&1&0&0\\30&1&1&1&1&0\\31&1&1&1&1&1\end{vmatrix}=\begin{vmatrix}0&0&&&1\\0&0&0&1&0\\0&1&0&0&1\\0&1&0&1&1\\0&1&1&0&1\\1&0&0&0&0\\1&0&0&0&1\\1&0&0&1&0\\1&0&1&1&0\\1&1&0&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}=\begin{vmatrix}0&0&&&1\\0&0&0&1&0\\0&1&0&0&1\\0&1&0&1&1\\0&1&1&0&1\\0&1&1&1&1\\1&0&0&0&0\\1&0&0&0&1\\1&0&0&1&0\\1&0&1&0&0\\1&0&1&1&0\\1&1&0&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}=\begin{vmatrix}0&0&&&1\\0&0&0&1&0\\0&1&&&1\\1&0&&&0\\1&0&0&0&1\\1&1&&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}=$$
$$=\begin{vmatrix}0&&&&1\\0&0&0&1&\\1&0&&&0\\1&0&0&0&\\1&&&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}=\begin{vmatrix}0&&&&1\\0&0&0&1&\\1&0&&0&0\\1&0&&1&0\\1&0&0&0&\\1&&&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}=\begin{vmatrix}0&&&&1\\0&0&0&1&\\1&0&0&0&0\\1&0&1&0&0\\1&0&0&0&\\1&&&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}=\begin{vmatrix}0&&&&1\\0&0&0&1&\\1&0&0&0&\\1&&&1&0\\&&1&0&0\\&1&1&1&\end{vmatrix}.$$

Fig. 1. Simplification of the Boolean function by duplicating the constituents and elementary conjunctions

Thus, a short analysis is required for the priority application of operations in the first binary matrix (to establish the optimal stack of logical operations). That ensures proper minimization efficiency for the earlier unaccounted-for variants of simplifying Boolean functions by the visual-matrix form of the analytical method [12]. For a series of cases, the choice of the optimal stack is also necessary for calculations in the second binary matrix.

6. Analysis of methods for simplifying a Boolean function of the general form

The equivalent transformations of the first matrix convert the PDNF of the assigned function into an arbitrary DNF. Effective simplification of the arbitrary DNF of the Boolean function is carried out with the help of logical operations discussed in p. 4, 5. It is possible that arbitrary DNF can be simplified by decoupling the parallel conjunct terms [22]. The function's DNF, in particular, is simplified using the Blake-Poretsky method [23].

Example 6: It is required to find the minimum algebraic form of the Boolean function $f(a, b, c, d, e)$ (12) [3, 24].

$$f(a,b,c,d,e)=ab+a\bar{c}d+bc+bde+\bar{a}de+\bar{c}de. \tag{12}$$

Solution:

$$f(a,b,c,d,e)=\left|\begin{array}{ccccc} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ & 1 & & 1 & 1 \\ 0 & & & 1 & 1 \\ & & 0 & 1 & 1 \end{array}\right|=\left|\begin{array}{ccccc} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ 0 & & & 1 & 1 \\ & & 0 & 1 & 1 \end{array}\right|=$$

For didactic convenience of figurative transformations, the right-hand matrix is rewritten to a new row since the current simplification procedure uses a common block from the previous logical operation:

$$=\left|\begin{array}{ccccc} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ 0 & & & 1 & 1 \\ & & 0 & 1 & 1 \end{array}\right|=\left|\begin{array}{ccccc} 1 & 1 & & & \\ 1 & & 0 & 1 & \\ & 1 & 1 & & \\ 0 & & & 1 & 1 \end{array}\right|=$$

$$=b(a+c)+d(a\bar{c}+\bar{a}e).$$

The result $f(a,b,c,d,e)=b(a+c)+d(a\bar{c}+\bar{a}e)$ contains eight literals. This is one literal less than in [3] and two literals less than in [24].

7. The results of Boolean function minimization by the visual-matrix form of the analytical method

A protocol with a relatively complex algorithm for applying absorption operation and the operation of super-gluing the variables (p. 4), a stack of logical operations in the first matrix (p. 5), defines a new standard, compared to [12], for minimizing Boolean functions. This improves the efficiency of the procedure, which makes it possible, in particular, to simplify logical functions with a relatively larger number of input variables manually.

Example 7: It is required to minimize the logical function $F(x_1, x_2, x_3, x_4, x_5)$ by the analytical method, which is represented in the canonical form [25].

$$F=\Sigma\,(0, 2, 5, 6, 7, 8, 10, 16, 19, 20, 23, 27, 31). \tag{13}$$

Solution:

$$F(x_1,x_2,x_3,x_4,x_5)=$$

$$=\left|\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \\ 6 & 0 & 0 & 1 & 1 & 0 \\ 7 & 0 & 0 & 1 & 1 & 1 \\ 8 & 0 & 1 & 0 & 0 & 0 \\ 10 & 0 & 1 & 0 & 1 & 0 \\ 16 & 1 & 0 & 0 & 0 & 0 \\ 19 & 1 & 0 & 0 & 1 & 1 \\ 20 & 1 & 0 & 1 & 0 & 0 \\ 23 & 1 & 0 & 1 & 1 & 1 \\ 27 & 1 & 1 & 0 & 1 & 1 \\ 31 & 1 & 1 & 1 & 1 & 1 \end{array}\right|=\left|\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & \\ 1 & 0 & & 0 & 0 \\ 1 & & & 1 & 1 \end{array}\right|=\left|\begin{array}{ccccc} 0 & & 0 & & 0 \\ 0 & 0 & 1 & & 1 \\ 0 & 0 & 1 & 1 & \\ 1 & 0 & & 0 & 0 \\ 1 & & & 1 & 1 \end{array}\right|. \tag{14}$$

Table 1 gives the results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$ (13) using the Karnaugh map, the Quine-McCluskey method, the method of undefined coefficients [25], and the analytical method.

Table 1

Result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$ (13)

Karnaugh map, Quine-Mc-Clusky method, the method of undefined coefficients	Analytical method
$F=\bar{x}_1\bar{x}_2x_3x_5+x_1\bar{x}_2\bar{x}_4\bar{x}_5+$ $+\bar{x}_1\bar{x}_3x_4\bar{x}_5+\bar{x}_1\bar{x}_2x_4\bar{x}_5+$ $+\bar{x}_1\bar{x}_3\bar{x}_4\bar{x}_5+x_1x_4x_5$	$F=\bar{x}_1\bar{x}_3\bar{x}_5+\bar{x}_1\bar{x}_2x_3x_5+$ $+\bar{x}_1\bar{x}_2x_3x_4+x_1\bar{x}_2\bar{x}_4\bar{x}_5+$ $+x_1x_4x_5$

Contemplating Table 1 reveals that the result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$ (14) by the analytical method is the minimum function containing five minterms. This is one minterm less than in [25].

The result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5)$ (13) using the software Logic Friday 1.1.4 [26] is as follows:

$$F=A'B'DE'+A'C'E'+ADE+AB'D'E'+A'B'CE. \tag{15}$$

The minimum function (15) contains 11 inverted variables. This is one inverted variable more than in (14).

Logic Friday provides a graphical interface for the Espresso program [27].

Example 8: It is required to minimize the logical function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ by the analytical method, which is represented in the canonical form [28].

$$F=\sum\begin{pmatrix} 0,\ 1,\ 2,\ 3,\ 4,\ 9,\ 10,\ 11{,}12,\ 15,\ 18,\ 19, \\ 20,\ 21,\ 22,\ 23,\ 26,\ 27,\ 28,\ 29,\ 32, \\ 33,\ 38,\ 39,\ 40,\ 41,\ 42,\ 43,\ 46,\ 47, \\ 48,\ 51,\ 52,\ 53,\ 54,\ 55,\ 56,\ 57,\ 62,\ 63 \end{pmatrix}. \tag{16}$$

Solution:

$$F(x_1,x_2,x_3,x_4,x_5,x_6)=$$

$$=\left|\begin{array}{r|cccccc||r|cccccc}
0 & 0&0&0&0&0&0 & 32 & 1&0&0&0&0&0\\
1 & 0&0&0&0&0&1 & 33 & 1&0&0&0&0&1\\
2 & 0&0&0&0&1&0 & 38 & 1&0&0&1&1&0\\
3 & 0&0&0&0&1&1 & 39 & 1&0&0&1&1&1\\
4 & 0&0&0&1&0&0 & 40 & 1&0&1&0&0&0\\
9 & 0&0&1&0&0&1 & 41 & 1&0&1&0&0&1\\
10 & 0&0&1&0&1&0 & 42 & 1&0&1&0&1&0\\
11 & 0&0&1&0&1&1 & 43 & 1&0&1&0&1&1\\
12 & 0&0&1&1&0&0 & 46 & 1&0&1&1&1&0\\
15 & 0&0&1&1&1&1 & 47 & 1&0&1&1&1&1\\
18 & 0&1&0&0&1&0 & 48 & 1&1&0&0&0&0\\
19 & 0&1&0&0&1&1 & 51 & 1&1&0&0&1&1\\
20 & 0&1&0&1&0&0 & 52 & 1&1&0&1&0&0\\
21 & 0&1&0&1&0&1 & 53 & 1&1&0&1&0&1\\
22 & 0&1&0&1&1&0 & 54 & 1&1&0&1&1&0\\
23 & 0&1&0&1&1&1 & 55 & 1&1&0&1&1&1\\
26 & 0&1&1&0&1&0 & 56 & 1&1&1&0&0&0\\
27 & 0&1&1&0&1&1 & 57 & 1&1&1&0&0&1\\
28 & 0&1&1&1&0&0 & 62 & 1&1&1&1&1&0\\
29 & 0&1&1&1&0&1 & 63 & 1&1&1&1&1&1
\end{array}\right|=\left|\begin{array}{cccccc}
0&0&0&0&&\\
0&0&&1&0&0\\
0&0&1&0&0&1\\
0&0&1&0&1&0\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&1&1&0&\\
1&0&0&0&0&\\
1&0&&1&1&\\
1&0&1&0&&\\
1&1&0&0&0&0\\
1&1&0&0&1&1\\
1&1&1&0&0&\\
1&1&1&1&1&
\end{array}\right|=$$

$$=\left|\begin{array}{cccccc}
0&0&0&0&&\\
0&&&1&0&0\\
0&0&1&0&&1\\
0&0&1&0&1&\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&0&&1&1&\\
1&0&1&0&&\\
1&&0&0&0&0\\
&1&0&0&1&1\\
1&&1&0&0&\\
1&1&&1&1&
\end{array}\right|=\left|\begin{array}{cccccc}
0&0&0&0&&\\
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&0&1&0&&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|=\left|\begin{array}{cccccc}
0&0&0&0&1&\\
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
0&0&1&&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|=$$

$$=\left|\begin{array}{cccccc}
0&0&&0&1&\\
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
0&0&1&0&1&1\\
0&0&1&1&1&1\\
0&1&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|=\left|\begin{array}{cccccc}
0&&&1&0&0\\
&0&1&0&&1\\
&0&1&0&1&\\
&0&1&&1&1\\
0&&&0&1&\\
&1&0&1&&\\
0&1&&1&0&\\
&0&0&0&0&\\
1&&&1&1&\\
1&&&0&0&0\\
&1&0&&1&1\\
1&&1&0&0&
\end{array}\right|. \qquad (17)$$

Table 2 gives the results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (16) using the software that employs ASCII files to import and export data and is based on the Quine-McClusky method [28] and the analytical method.

Table 2

Result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (16)

Software based on the Quine-Mc-Cluskey method [28]	Analytical method
$F=\bar{x}_1x_4\bar{x}_5\bar{x}_6+\bar{x}_2x_3x_5x_6+\bar{x}_1x_2x_4\bar{x}_5+\bar{x}_1\bar{x}_2\bar{x}_4x_6+\bar{x}_1\bar{x}_4x_5+x_2\bar{x}_3x_4+\bar{x}_2x_3\bar{x}_4x_5+\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5+x_1x_4x_5+x_1\bar{x}_4\bar{x}_5\bar{x}_6+x_2\bar{x}_3x_5x_6+x_1x_3\bar{x}_4\bar{x}_5$	$F=\bar{x}_1x_4\bar{x}_5\bar{x}_6+\bar{x}_2x_3\bar{x}_4x_6+\bar{x}_2x_3\bar{x}_4x_5+\bar{x}_2x_3x_5x_6+\bar{x}_1\bar{x}_4x_5+x_2\bar{x}_3x_4+\bar{x}_1x_2x_4\bar{x}_5+\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5+x_1x_4x_5+x_1\bar{x}_4\bar{x}_5\bar{x}_6+x_2\bar{x}_3x_5x_6+x_1x_3\bar{x}_4\bar{x}_5$

Table 2 demonstrates that the result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (16) by the analytical method is a minimal function containing twenty-three inverted variables. This is one inverted variable less than in [28].

The results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (16) using the software JQM–Java Quine-McClusky, 1.2.4 [29] and the analytical method match.

The result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (16) using the software Logic Friday 1.1.4 [26] is as follows:

$$F=AD'E'F'+B'CD'E+B'D'E'F+B'C'D'E'+$$
$$+A'DE'F'+A'BDE'+ACD'E'+BC'D+ADE+$$
$$+B'CEF+A'D'E+BC'EF. \qquad (18)$$

The minimum function (18) contains 24 inverted variables. This is one inverted variable more than in (17).

Example 9: It is required to minimize the logical function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ by the analytical method assigned in DNF (Table 3) [25].

Table 3

Logical function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ truth table [25]

x_1	x_2	x_3	x_4	x_5	x_6	x_1	x_2	x_3	x_4	x_5	x_6
0	0	1	–	0	1	0	1	1	0	0	1
0	1	0	–	1	0	0	1	1	0	1	1
0	1	0	–	1	1	0	1	1	1	0	0
0	1	1	–	0	1	0	1	1	1	1	1
1	0	0	–	1	1	1	0	0	1	1	0
1	0	1	–	0	1	1	0	0	1	1	1
1	1	0	–	0	1	1	0	1	0	1	1
1	1	0	–	1	0	1	0	1	1	0	1
1	1	1	–	0	0	1	0	1	1	1	0
0	0	0	0	0	0	1	0	1	1	1	1
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	1	1	1	1	0	0	1	1
0	0	1	1	1	0	1	1	0	1	0	0
0	0	1	1	1	1	1	1	0	1	0	1
0	1	0	0	0	1	1	1	0	1	1	1
0	1	0	1	0	1	1	1	1	0	0	1
0	1	0	1	1	1	1	1	1	0	1	0
0	1	1	0	0	0	1	1	1	0	1	1

Solution:

$$F(x_1,x_2,x_3,x_4,x_5,x_6)=$$

$$
=\left|\begin{array}{cccccc}
0&0&1&&0&1\\
0&1&0&&1&0\\
0&1&0&&1&1\\
0&1&1&&0&1\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&0\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
0&0&0&1&1&0\\
0&0&1&0&1&1\\
0&0&1&1&1&0\\
0&0&1&1&1&1\\
0&1&0&0&0&1\\
0&1&0&1&0&1\\
0&1&0&1&1&1\\
0&1&1&0&0&0\\
0&1&1&0&0&1\\
0&1&1&0&1&1\\
0&1&1&1&0&0\\
0&1&1&1&1&1\\
1&0&0&1&1&0\\
1&0&0&1&1&1\\
1&0&1&0&1&1\\
1&0&1&1&0&1\\
1&0&1&1&1&0\\
1&0&1&1&1&1\\
1&1&0&0&0&1\\
1&1&0&0&1&1\\
1&1&0&1&0&0\\
1&1&0&1&0&1\\
1&1&0&1&1&1\\
1&1&1&0&0&1\\
1&1&1&0&1&0\\
1&1&1&0&1&1
\end{array}\right|
=\left|\begin{array}{cccccc}
0&0&1&&0&1\\
0&1&0&&1&0\\
0&1&0&&1&1\\
0&1&1&&0&1\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&0\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
0&0&0&1&1&0\\
0&0&1&0&1&1\\
0&0&1&1&1&0\\
0&0&1&1&1&1\\
0&1&0&0&0&1\\
0&1&0&1&0&1\\
0&1&1&0&0&0\\
0&1&1&0&1&1\\
0&1&1&1&0&0\\
0&1&1&1&1&1\\
1&0&0&1&1&0\\
1&0&1&0&1&1\\
1&0&1&1&1&0\\
1&0&1&1&1&1\\
1&1&0&0&1&1\\
1&1&0&1&0&0\\
1&1&0&1&0&1\\
1&1&0&1&1&1\\
1&1&1&0&0&1\\
1&1&1&0&1&0\\
1&1&1&0&1&1\\
1&1&0&1&1&0\\
1&1&1&0&0&0
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&0&1\\
0&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&0\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
0&0&0&1&1&0\\
0&0&1&0&1&1\\
0&0&1&1&1&0\\
0&0&1&1&1&1\\
0&1&0&0&0&1\\
0&1&0&1&0&1\\
0&1&1&0&0&0\\
0&1&1&0&1&1\\
0&1&1&1&0&0\\
0&1&1&1&1&1\\
1&0&0&1&1&0\\
1&0&1&0&1&1\\
1&0&1&1&1&0\\
1&0&1&1&1&1\\
1&1&0&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&0&1\\
0&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&0&&1&\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&0&1&1\\
&0&1&1&1&\\
0&1&0&&0&1\\
0&1&1&&0&0\\
0&1&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&0&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
1&1&0&&0&1\\
1&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&&1&1\\
&0&1&1&1&\\
0&1&0&&0&1\\
0&1&1&&0&0\\
0&1&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&0&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&&1&1\\
&0&1&1&1&\\
0&&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|=
$$

$$
=\left|\begin{array}{cccccc}
0&&1&&&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&0&1&1&0\\
&0&1&&1&1\\
&0&1&1&1&0\\
&0&1&1&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
&0&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&&1\\
&1&0&&1&\\
1&0&0&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
0&0&1&&1&1\\
1&0&1&&1&1\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&&1\\
&1&0&&1&\\
1&0&&&1&1\\
1&0&1&&0&1\\
&1&0&&0&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&&1\\
&1&0&&1&\\
1&0&&&1&1\\
1&0&1&&&1\\
&1&0&&&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|
=\left|\begin{array}{cccccc}
0&&1&&&1\\
&1&0&&1&\\
1&0&&&1&1\\
&0&1&&&1\\
&1&0&&&1\\
&1&1&&0&0\\
0&0&0&0&0&0\\
&0&&1&1&0\\
1&1&0&1&&\\
1&1&1&0&&
\end{array}\right|.
$$

Table 4 gives the results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (Table 3) using a Karnaugh map, a Quine-McClusky method, the method of undefined coefficients [25], and the analytical method.

Table 4

Result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (Table 3) [25]

Karnaugh map, Quine-McClusky method, method of undefined coefficients	Analytical method
$F=\bar{x}_1x_2x_6+x_2\bar{x}_3x_5+x_1\bar{x}_2x_5x_6+\bar{x}_2x_3x_6+x_2\bar{x}_3x_6+x_2x_3\bar{x}_4\bar{x}_5+x_2x_3\bar{x}_5\bar{x}_6+\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5\bar{x}_6+\bar{x}_2x_4x_5\bar{x}_6+x_1x_2\bar{x}_3x_4+x_1x_2\bar{x}_4x_5$	$F=\bar{x}_1x_3x_6+x_2\bar{x}_3x_5+x_1\bar{x}_2x_5x_6+\bar{x}_2x_3x_6+x_2\bar{x}_3x_6+x_2x_3\bar{x}_5\bar{x}_6+\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5\bar{x}_6+\bar{x}_2x_4x_5\bar{x}_6+x_1x_2\bar{x}_3x_4+x_1x_2x_3\bar{x}_4$

Contemplating Table 4 reveals that the result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6)$ (Table 3) using the analytical method is the minimum function containing ten minterms. This is one minterm less than in [25].

Example 10: It is required to minimize the logical function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ by the analytical method, which is represented in the canonical form [30]:

$$F=\sum\begin{pmatrix}1,\ 3,\ 5,\ 7,\ 20,\ 21,\ 33,\ 61,\ 63,\ 73,\ 75,\ 77,\\ 79,\ 81,\ 83,\ 85,\ 96,\ 97,\ 99,\ 124,\ 125,\ 127\end{pmatrix}. \quad (19)$$

Solution:

$$F(A,B,C,D,E,F,G)=$$

$$=\begin{vmatrix} 1 & 0&0&0&0&0&0&1\\ 3 & 0&0&0&0&0&1&1\\ 5 & 0&0&0&0&1&0&1\\ 7 & 0&0&0&0&1&1&1\\ 20 & 0&0&1&0&1&0&0\\ 21 & 0&0&1&0&1&0&1\\ 33 & 0&1&0&0&0&0&1\\ 61 & 0&1&1&1&1&0&1\\ 63 & 0&1&1&1&1&1&1\\ 73 & 1&0&0&1&0&0&1\\ 75 & 1&0&0&1&0&1&1\\ 77 & 1&0&0&1&1&0&1\\ 79 & 1&0&0&1&1&1&1\\ 81 & 1&0&1&0&0&0&1\\ 83 & 1&0&1&0&0&1&1\\ 85 & 1&0&1&0&1&0&1\\ 96 & 1&1&0&0&0&0&0\\ 97 & 1&1&0&0&0&0&1\\ 99 & 1&1&0&0&0&1&1\\ 124 & 1&1&1&1&1&0&0\\ 125 & 1&1&1&1&1&0&1\\ 127 & 1&1&1&1&1&1&1 \end{vmatrix}=\begin{vmatrix} 0&0&0&0&&&1\\ 0&0&1&0&1&0&\\ 0&1&0&0&0&0&1\\ &1&1&1&1&&1\\ 1&0&0&1&&&1\\ 1&0&1&0&0&&1\\ 1&0&1&0&1&0&1\\ 1&1&0&0&0&0&0\\ 1&1&0&0&0&&1\\ 1&1&1&1&1&0&0 \end{vmatrix}=\begin{vmatrix} 0&0&0&0&&&1\\ 0&0&1&0&1&0&\\ 0&1&0&0&0&0&1\\ &1&1&1&1&&1\\ 1&0&0&1&&&1\\ 1&0&1&0&0&&1\\ 1&0&1&0&&0&1\\ 1&1&0&0&0&0&\\ 1&1&0&0&0&&1\\ 1&1&1&1&1&0& \end{vmatrix}=\begin{vmatrix} 0&0&0&0&&&1\\ 0&0&1&0&1&0&\\ &1&0&0&0&0&1\\ &1&1&1&1&&1\\ 1&0&0&1&&&1\\ 1&0&1&0&0&&1\\ 1&0&1&0&&0&1\\ 1&1&0&0&0&0&\\ 1&1&0&0&0&&1\\ 1&1&1&1&1&0& \end{vmatrix}.$$

Table 5 gives the results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (19) using a Karnaugh map [30] and the analytical method.

Table 5

Result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (19)

Tabular method (Karnaugh map)	Analytical method
$F=\bar{A}\bar{B}\bar{C}\bar{D}G+\bar{A}\bar{B}C\bar{D}\bar{E}\bar{F}+\bar{A}B\bar{C}\bar{D}\bar{E}\bar{F}G+BCDEG+A\bar{B}\bar{C}DG+A\bar{B}C\bar{D}\bar{E}G+A\bar{B}C\bar{D}\bar{F}G+ABC\bar{D}\bar{E}\bar{F}+ABC\bar{D}\bar{E}G+ABCDE\bar{F}$	$F=\bar{A}\bar{B}\bar{C}\bar{D}G+\bar{A}\bar{B}C\bar{D}E\bar{F}+A\bar{C}\bar{D}\bar{E}\bar{F}G+BCDEG+A\bar{B}\bar{C}DG+A\bar{B}C\bar{D}\bar{E}G+A\bar{B}C\bar{D}\bar{F}G+ABC\bar{D}\bar{E}\bar{F}+ABC\bar{D}\bar{E}G+ABCDE\bar{F}$

Table 5 demonstrates that the result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (19) by the analytical method is the minimum function containing 57 literals. This is one literal less than in [30].

The results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (19) using the software Logic Friday 1.1.4 [26] and the analytical method match.

Example 11: It is required to minimize the logical function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ by the analytical method, which is assigned in the canonical form [30]:

$$F=\sum\begin{pmatrix}1,\ 3,\ 5,\ 7,\ 8,\ 9,\ 10,\ 18,\ 20,\ 21,\ 23,\ 24,\\ 29,\ 31,\ 33,\ 35,\ 37,\ 39,\ 40,\ 43,\ 48,\ 51,\\ 53,\ 55,\ 57,\ 59,\ 61,\ 63,\ 64,\ 65,\ 66,\ 69,\\ 73,\ 77,\ 79,\ 81,\ 83,\ 84,\ 85,\ 89,\ 91,\ 93,\\ 97,\ 99,\ 101,\ 102,\ 103,\ 105,\ 106,\ 108,\\ 113,\ 115,\ 116,\ 118,\ 121,\ 123,\ 124,\ 127\end{pmatrix}. \quad (20)$$

Note: the values in Σ are the minterms for the rows of the truth table when the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ returns «1» at the output.

Solution:

Define the stack of logical operations of the first matrix of the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (20) as follows. By constructing the truth table of function (20) and calculating the number of unities and zeros separately in each column of the constructed table (according to the algorithm given in [12]), it can be established that the largest number of unities is in the extreme right column. The next step is to combine the sets of variables that contain a unity in the far-right position of the set into a separate matrix. Such a matrix is given in (21) as the first one. In another separate matrix, combine the sets of variables for function (20), which contain zeros in the far-right position of the set. Such a matrix is given in (22) as the first one. The simplification of function (20) in each matrix is carried out separately.

The matrix with the same unities in the extreme right column and its simplification takes the following form (21):

$$\left|\begin{array}{c|ccccccc||c|ccccccc}
1 & 0&0&0&0&0&0&1 & 65 & 1&0&0&0&0&0&1\\
3 & 0&0&0&0&0&1&1 & 69 & 1&0&0&0&1&0&1\\
5 & 0&0&0&0&1&0&1 & 73 & 1&0&0&1&0&0&1\\
7 & 0&0&0&0&1&1&1 & 77 & 1&0&0&1&1&0&1\\
9 & 0&0&0&1&0&0&1 & 79 & 1&0&0&1&1&1&1\\
21 & 0&0&1&0&1&0&1 & 81 & 1&0&1&0&0&0&1\\
23 & 0&0&1&0&1&1&1 & 83 & 1&0&1&0&0&1&1\\
29 & 0&0&1&1&1&0&1 & 85 & 1&0&1&0&1&0&1\\
31 & 0&0&1&1&1&1&1 & 89 & 1&0&1&1&0&0&1\\
33 & 0&1&0&0&0&0&1 & 91 & 1&0&1&1&0&1&1\\
35 & 0&1&0&0&0&1&1 & 93 & 1&0&1&1&1&0&1\\
37 & 0&1&0&0&1&0&1 & 97 & 1&1&0&0&0&0&1\\
39 & 0&1&0&0&1&1&1 & 99 & 1&1&0&0&0&1&1\\
43 & 0&1&0&1&0&1&1 & 101 & 1&1&0&0&1&0&1\\
51 & 0&1&1&0&0&1&1 & 103 & 1&1&0&0&1&1&1\\
53 & 0&1&1&0&1&0&1 & 105 & 1&1&0&1&0&0&1\\
55 & 0&1&1&0&1&1&1 & 113 & 1&1&1&0&0&0&1\\
57 & 0&1&1&1&0&0&1 & 115 & 1&1&1&0&0&1&1\\
59 & 0&1&1&1&0&1&1 & 121 & 1&1&1&1&0&0&1\\
61 & 0&1&1&1&1&0&1 & 123 & 1&1&1&1&0&1&1\\
63 & 0&1&1&1&1&1&1 & 127 & 1&1&1&1&1&1&1
\end{array}\right| = \left|\begin{array}{cccccc}
0& &0&0& & \\
0&0&0&1&0&0\\
0&0&1& &1& \\
0&1&0&1&0&1\\
0&1&1&0&0&1\\
0&1&1&0&1& \\
0&1&1&1& & \\
1&0&0& & &0\\
1&0&0&1&1&1\\
1& &1& &0& \\
1&0&1& &1&0\\
1&1&0&0& & \\
1&1&0&1&0&0\\
1&1&1&1&1&1
\end{array}\right| =$$

$$= \left|\begin{array}{cccccc}
0& &0&0& & \\
0&0&0&1&0&0\\
0&0&1& &1& \\
0&1& &1&0&1\\
0&1&1& &0&1\\
0&1&1& &1& \\
0&1&1&1& & \\
1&0&0& & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
1&0&1& & &0\\
1&1&0&0& & \\
1&1&0& &0&0\\
 &1&1&1& &1
\end{array}\right| = \left|\begin{array}{cccccc}
0& &0&0& & \\
0&0&0& &0&0\\
0& &1& &1& \\
0&1& &1&0&1\\
0&1&1& &0&1\\
0&1&1&1& & \\
1&0& & & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
 &1&0&0& & \\
1&1&0& &0&0\\
 &1&1&1& &1
\end{array}\right| = \left|\begin{array}{cccccc}
0& &0&0& & \\
 &0&0& &0&0\\
0& &1& &1& \\
0&1& &1&0&1\\
0&1&1& &0&1\\
0&1&1&1& & \\
1&0& & & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
 &1&0&0& & \\
1& &0& &0&0\\
 &1&1&1& &1
\end{array}\right| = \left|\begin{array}{cccccc}
0& &0&0& & \\
 &0&0& &0&0\\
0& &1& &1& \\
0&1& &1&0&1\\
0&1&1& &0&1\\
0&1&1&1& & \\
1&0& & & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
 &1&0&0& & \\
1& & & &0&0\\
 &1&1&1& &1
\end{array}\right| =$$

$$= \left|\begin{array}{cccccc}
0& &0&0& & \\
 &0&0& &0&0\\
0& &1& &1& \\
0&1&0&1&0&1\\
0&1&1&1&0&1\\
0&1&1& &0&1\\
0&1&1&1& & \\
1&0& & & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
 &1&0&0& & \\
1& & & &0&0\\
 &1&1&1& &1
\end{array}\right| = \left|\begin{array}{cccccc}
0& &0&0& & \\
 &0&0& &0&0\\
0& &1& &1& \\
0&1&0& &0&1\\
0&1&1& &0&1\\
0&1&1&1& & \\
1&0& & & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
 &1&0&0& & \\
1& & & &0&0\\
 &1&1&1& &1
\end{array}\right| = \left|\begin{array}{cccccc}
0& &0&0& & \\
 &0&0& &0&0\\
0& &1& &1& \\
0&1& & &0&1\\
0&1&1&1& & \\
1&0& & & &0\\
1&0&0&1&1& \\
1& &1& &0& \\
 &1&0&0& & \\
1& & & &0&0\\
 &1&1&1& &1
\end{array}\right|. \qquad (21)$$

The matrix with the same zeros in the extreme right column and its simplification is as follows (22):

$$\left|\begin{array}{c|ccccccc}
8 & 0&0&0&1&0&0&0\\
10 & 0&0&0&1&0&1&0\\
18 & 0&0&1&0&0&1&0\\
20 & 0&0&1&0&1&0&0\\
24 & 0&0&1&1&0&0&0\\
40 & 0&1&0&1&0&0&0\\
48 & 0&1&1&0&0&0&0\\
64 & 1&0&0&0&0&0&0\\
66 & 1&0&0&0&0&1&0\\
84 & 1&0&1&0&1&0&0\\
102 & 1&1&0&0&1&1&0\\
106 & 1&1&0&1&0&1&0\\
108 & 1&1&0&1&1&0&0\\
116 & 1&1&1&0&1&0&0\\
118 & 1&1&1&0&1&1&0\\
124 & 1&1&1&1&1&0&0
\end{array}\right| =$$

$$= \left|\begin{array}{cccccc}
0&0&0&1&0& \\
0&0&1&0&0&1\\
 &0&1&0&1&0\\
0&0&1&1&0&0\\
0&1&0&1&0&0\\
0&1&1&0&0&0\\
1&0&0&0&0& \\
1&1&0&0&1&1\\
1&1&0&1&0&1\\
1&1& &1&1&0\\
1&1&1&0&1&
\end{array}\right| =$$

$$= \left|\begin{array}{cccccc}
0&0&0&1&0& \\
0&0&1&0&0&1\\
 &0&1&0&1&0\\
0&0& &1&0&0\\
0& &0&1&0&0\\
0&1&1&0&0&0\\
1&0&0&0&0& \\
1&1& &0&1&1\\
1&1&0&1&0&1\\
1&1& &1&1&0\\
1&1&1&0&1&
\end{array}\right|. \qquad (22)$$

By combining the results of minimization of two matrices, taking into consideration the bits of the extreme right column of the first matrices in (21) and (22) one obtains MDNF (23) of the assigned Boolean function (20).

$$F_{\mathrm{MDNF}} = $$

$$= \left|\begin{array}{ccccccc} 0 & & 0 & 0 & & & 1 \\ & 0 & 0 & & 0 & 0 & 1 \\ 0 & & 1 & & 1 & & 1 \\ 0 & 1 & & & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & & & 1 \\ 1 & 0 & & & & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & & 1 \\ 1 & & 1 & & 0 & & 1 \\ & 1 & 0 & 0 & & & 1 \\ 1 & & & & 0 & 0 & 1 \\ & 1 & 1 & 1 & & 1 & 1 \end{array}\right| . \left|\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & & 1 & 0 & 0 & 0 \\ 0 & & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & & 0 \\ 1 & 1 & & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & & 0 \end{array}\right| . \quad (23)$$

The minimum function (23) consists of 22 minterms and contains 118 literals.

The result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (20) using a Karnaugh map [30] is as follows:

$$\begin{aligned} F=&A'C'D'G+A'B'C'DE'F'+\\ &+A'B'C'DE'G'+A'B'CD'E'FG'+\\ &+B'CD'EF'+A'B'CEG+\\ &+A'B'CDE'F'G'+A'BCG+\\ &+A'BC'DE'G'+AB'C'D'E'G'+\\ &+AB'F'G+AB'C'DEG+\\ &+ACE'F+ABCEF'G'+ABCD'EG'+\\ &+ABCDFG+BC'D'G+ABC'D'EF+\\ &+ABC'E'F'G+ABC'DEF'G'+\\ &+ABC'DE'FG'. \end{aligned} \quad (24)$$

However, when tested by code No. 43 – 0101011, function (24) does not yield unity. Given this, an error would be made in the expression of the minimum function (24) [30].

The result of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (20) using the Python software [31] is as follows:

$$\begin{aligned} F=&A'B'C'DE'G'+A'B'CD'E'FG'+\\ &+B'CD'EF'+A'B'DE'F'G'+\\ &+A'C'DE'F'G'+A'BE'FG+\\ &+A'BCD'E'F'G'+AB'C'D'E'G'+\\ &+AB'C'DEG+ABC'DE'FG'+\\ &+ABDEF'G'+BCDFG+C'D'F'G+\\ &+ACE'G+A'CEG+A'C'D'G+\\ &+B'C'E'F'G+BC'D'G+ABD'EFG'+\\ &+A'BCDG+AB'F'G+AE'F'G+\\ &+ACD'EF'G'. \end{aligned} \quad (25)$$

The minimum function (25) contains 23 minterms, which is one minterm more than the minimum function derived by the analytical method (23).

The results of minimizing the function $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ (20) using the software Logic Friday 1.1.4 [26] and the analytical method match.

Example 12. It is required to minimize a 64-bit Boolean function represented in PDNF by the analytical method (Table 6) [28].

Table 6

Truth table for the 64-bit PDNF of logical function [28]

0000000000011101101001110010010100000000010011100011011101111 0101
0000000000011101101001110010110100000000010011100011011101111 0101
0000000000011101101001110011010100000000010011100011011101111 0101
0000000000011101101001110011110100000000010011100011011101111 0101
0000000000100011110101011101001000000010010000000010111100110011
0000000000100011110101011101001001000010010000000010111100110011
0000000000100011110101011101001010000010010000000010111100110011
0000000000100011110101011101001011000010010000000010111100110011
0000000010011100011011101111010100000000110011110011111010000 1100
0000000010011100011011101111010100001001100111100111111100001100
0000000010011100011011101111010100010001100111100111110100001100
0000000010011100011011101111010100011001100111100111110100001100
0000000010100010001111000111001000000000110100001001000011101 0011
0000000010100010001111000111001000000000110100001001000011101 1011
0000000010100010001111000111001000000000110100001001000011111 0011
0000000010100010000111100011100100000000011010000100100001111 10011
0000000010100010001111000111001000000000110100001001000011111 1011
0000000010100010101111000111001000000000110100001001000011111 0011
0000000010100011001111000111001000000000110100001001000011111 0011
0000000010100011101111000111001000000000110100001001000011111 0011
0000000100100000000101111001100110000000101000100011110001110010
0000000100100000000101111001100110100000101000100011110001110010
0000000100100000000101111001100111000000101000100011110001110010
0000000100100000000101111001100111100000101000100011110001110010
0000000101111111100001001101111010000000011111100100110110111 0100
0000000101111111100001001101111010010000011111100100110110111 0100
0000000101111111100001001101111010100000011111100100110110111 0100
0000000101111111100001001101111010110000011111100100110110111 0100
0000000110000000101000010101000110000000000100001101010001011 00010
0000000110000000101000010101000110000000010100001101010001011 00010
0000000110000000101000010101000110000000100100001101010001011 00010
0000000110000000101000010101000110000000110100001101010001011 00010
0000000110111111000000000011011100000000000011111010001001000111101
0000000110111111000000000011011100001000000011111010001001000111101
0000000110111111000000000011011100010000000011111010001001000111101
0000000110111111000000000011011100011000000011111010001001000111101
0000000111101110100001000000000110000000000000001110010110100000 10
0000000111101110100001000000000110010000000000001110010110100000 10
0000000111101110100001000000000110100000000000001110010110100000 10
0000000111101110100001000000000110110000000000001110010110100000 10
0000011000000111010010101100110000000001110000110001001111010 11001
0000011000000111011010101100110000000001110000110001001111010 11001
0000011000000111110010101100110000000001110000110001001111010 11001
0000011000000111111010101100110000000001110000110001001111010 11001
0000001100110011100100000000000001000000111011001001011001010000000
0000011001110101000000101100100100000110011101110001000100101000
0000011001110101001000101100100100000110011101110001000100101000
0000011001110101010000101100100100000110011101110001000100101000
0000011001110101011000101100100100000110011101110001000100101000
0000001101101011011101011001100000000011111010101111111000011 10001
0000001110111101001000101110101001000000110111101101001100011 101000
0001011011010110111010110011000000000111110101011111110000111 0001
0001011101111010010001011101010010000001101111011010011000111 01000
0010011000110111111100110111100000000011100110100000011010000 1001
0010011000110111111100110111100000100011100110100000011010000 1001
0010011000110111111100110111100001000011100110100000011010000 1001
0010011000110111111100110111100001100011100110100000011010000 1001
0010011011010110111010110011000000000111110101011111110000111 0001
0010011101111010010001011101010010000001101111011010011000111 01000
0011011011010110111010110011000000000111110101011111110000111 0001
0011011101111010010001011101010010000001101111011010011000111 01000
0100011001100111001000000000000010000011101100100101100101000 00000
1000011001100111001000000000000010000011101100100101100101000 00000
1100011001100111001000000000000010000011101100100101100101000 00000

In Table 6, color highlights complete binary combinatorial 2-(n, b)-design systems: 15 systems of 2-(2, 4)-design, and 1 system of 2-(1, 2)-design. To minimize the function, one needs to carry out 15 logical operations of super-gluing the variables and one operation of simply gluing the variables. After that, one needs to perform two more consecutive operations of semi-gluing the variables, first in the third row at the top, and then in the fourth row from above; thus, the minimum function (26) is obtained:

$$\begin{array}{l}
000000000001110110100111001\sim\sim1010000000001001110001101110111110101\\
000000000010001111010101110100100\sim\sim00001001000000010111100110011\\
00000000100111000110111011110101000\sim00011001111001111101000 01100\\
000000001001110001101110111101010000100110011110011111\sim100001100\\
000000001001110001101110111101010001\sim001100111100111110100001100\\
0000000010100010001111000111001000000001101000010010000111\sim1\sim011\\
000000001010001\sim\sim011110001110010000000001101000010010000111110011\\
0000000100100000001011110011001100\sim\sim0000101000100011110001110010\\
00000001011111111000010011011111010\sim\sim00000111111001001101101110100\\
000000011000000010100001010100011000000\sim\sim010000110101000101100010\\
0000000110111111000000000110111000\sim\sim00000011111010001001000111101\\
0000000111101110100001000000000110\sim\sim00000000000111001011010000010\\
000001100000111\sim1\sim01010110011000000000111000011000100111101011001\\
\sim\sim000110011001110010000000000000100000111011001001011001010000000\\
00000110011101010\sim\sim000101100100100000110011101110001000100101000\\
00\sim\sim0110110101101110101100110000000001111101010111111100001110001\\
00\sim\sim011101110100100010111010100100000110111101101001100011101000\\
001001100011011111110011011110000\sim\sim0001110011010000000110100001001
\end{array} \quad (26)$$

The results of the simplification of the function (Table 6) using software that uses ASCII files to import and export data and is based on the Quine-McCluskey method [28] and the analytical method are the same in the number of minterms. However, in [28], the fourth line at the top of the minimum function does not indicate a logical operation over the variable, so one must assume that the minimum function (26) contains one literal less.

8. Discussion of the results of Boolean function minimization by the visual-matrix form of the analytical method

A mathematical apparatus of the visual-matrix form of the analytical method is considered in works [8–10, 12, 32, 33] and others. The components of the method are given in Table 7.

Table 7

Technology of the analytical method for the visual-matrix form

1	Binary combinatorial systems with repeated 2-(n, b)-design, 2-(n, x/b)-design
2	Verbal and figurative representation of information
3	Logical operation of super-gluing the variables
4	Logical operation of incomplete super-gluing the variables
5	Hermeneutics of logical operations on binary structures
6	Protocols of figurative transformations
7	Attribute of the minimum logical function
8	Minimization of Boolean functions on the complete truth table
9	Algorithm of analytical method and its automation
10	Extension of the analytical method to other logical bases

This work has added new components to the technology for minimizing Boolean functions by the visual-matrix form of the analytical method (Table 8).

When simplifying logical functions, it is not always obvious which of the laws from the algebra of logic should be applied at a specific step. The visual structures of binary matrices and the unification of original procedures make it possible, to some extent, to resolve this issue.

The special feature of the visual-matrix form of the analytical method is that the method is based on the binary combinatorial repeated 2-(n, b)-design, 2-(n, x/b)-design systems. The truth table of logical functions is also a repeated combinatorial system. That makes it possible, during function minimization, to do without auxiliary objects such as Karnaugh maps, Mahony maps, Weich diagrams, acyclic graph, non-directed graph, cover tables, cubes, etc. The visualization of 2-dimensional binary matrices makes it possible to manually simplify the Boolean functions (using a mathematical editor, for example, MathType 7.4.0) within the limits of up to 64 input variables (example 12) for the PDNF (PKNF) representation of the function.

Table 8

New components in the technology of minimization by the visual-matrix form of the analytical method

1	Relatively complex algorithms for the use of logical absorption operation and the operation of super-gluing the variables
2	Stack of logical operations

Table 9 gives the results of Boolean function minimization borrowed from works by other authors and by the analytical method.

Table 9 demonstrates that the results of minimizing by the analytical method, JQM software – Java Quine-McCluskey 1.2.4 [29], and, in some cases, Logic Friday 1.1.4 [26], are the same. For other examples, the minimization by an analytical method demonstrates a minimal logical function, either less by one minterm, or less by one literal, or a function that contains less than one inverted variable.

Limiting the use of the visual-matrix form of the analytical method are those cases when the switch function is represented by a mixed basis. In this case, the function must be represented by one logical basis.

The weak side of the method considered is in the small practical application of the visual-matrix form of the analytical method to minimize Boolean functions, followed by the design and manufacture of the corresponding computational components. The negative internal factors of the method are associated with additional time costs for establishing protocols for simplifying logical functions, followed by the creation of a library of protocols that may illustrate the corresponding figurative transformations.

Table 9

Comparative table of the examples of Boolean function minimization borrowed from works by other authors and by the visual-matrix form of the analytical method

Example No.	Minimization method name	Number of input variables	Minimization result	Analytical method result
7	Karnaugh map, Quine-McCluskey method, the method of undefined coefficients [25]	5	6 minterms	5 minterms
7	Logic Friday 1.1.4 [26]	5	11 inverted variables	10 inverted variables
8	Software that uses ASCII files to import and export data and is based on the Quine-McCluskey method [28]	6	24 inverted variables	23 inverted variables
8	Software JQM – Java Quine-McCluskey 1.2.4 [29]	6	12 minterms	12 minterms
8	Logic Friday 1.1.4 [26]	6	24 inverted variables	23 inverted variables
9	Karnaugh map, Quine-McCluskey method, the method of undefined coefficients [25]	6	11 minterms	10 minterms
10	Karnaugh map [30]	7	58 literals	57 literals
10	Logic Friday 1.1.4 [26]	7	The results of minimization coincide	
11	Karnaugh map [30]	7	21 minterms (verification failed)	22 minterms
11	Software Python [31]	7	23 minterms	22 minterms
11	Logic Friday 1.1.4 [26]	7	The results of minimization coincide	
12	Software that uses ASCII files to import and export data and is based on the Quine-McCluskey method [28]	64	1,120 literals	1,119 literals

The prospect of further research may be the search for new rules for converting logical functions based on the Hamming distance.

9. Conclusions

1. Comparatively more complex algorithms for the use of logical operations of absorption and super-gluing the variables expand variants for their use, which makes it possible to improve the efficiency of the procedure for minimizing Boolean functions.

2. The optimal solution for minimizing Boolean functions by the analytical method is based on the primary application of the operation of super-gluing the variables (if such an operation is possible) within the truth table of the assigned Boolean function [12]. However, there are a small number of exceptions when minimization is more effective if one first applies the logical operation of simply gluing the variables [12]. Thus, a brief analysis is required for the primary use of logical operations in the first binary matrix. Such analysis is carried out by selecting, according to the established criteria, the optimal stack of logical operations for the first binary matrix, and, in some cases, for the second binary matrix (Example 5). This ensures the proper efficiency of minimization for the earlier unaccounted-for variants for simplifying Boolean functions by the analytical method.

3. After simplifying PDNF (PKNF) in the first binary matrix, the assigned function takes the form of DNF (KNF) representation, whereby becoming a function of the general form. The classic algorithm for minimizing the Boolean functions of the general form is the Blake-Poretsky method. However, in most cases, it would suffice to carry out 1–3 logical operations of the generalized gluing of variables. The minimization of logical functions of the general form is also possible when the operation of absorption of variables is applied. The visual-matrix form adjusts to a semi-intuitive approach to minimizing Boolean functions, based on some properties in the perception of binary matrices. Such methods are needed to manually minimize Boolean functions [3].

4. The effectiveness of the visual-matrix form of the analytical method to minimize Boolean functions is demonstrated by the following examples:

– example 3 [13] – the minimization of a 4-bit Boolean function;

– example 4 [21] – the minimization of a 3-bit Boolean function;

– example 5 [12], example 7 [25] – the minimization of 5-bit Boolean functions;

– example 8 [28], example 9 [25] – the minimization of 6-bit Boolean functions;

– example 10 [30], example 11 [30] – the minimization of 7-bit Boolean functions;

– example 12 [28] – the minimization of a 64-bit Boolean function.

The results of the comparison have established that the effectiveness of the visual-matrix form of the analytical method for minimizing Boolean functions gives grounds for the feasibility of its application in the procedures for minimizing logical functions since the visual-matrix form of the analytical method allows the following:

– ensure the prompt selection of the stack of logical operations in the first and second binary matrices, which ultimately gives the optimal scenario for minimizing logical functions;

– improve the efficiency of the procedure for minimizing logical functions by implementing relatively complex algorithms for the use of logical operations of absorption and super-gluing the variables.

References

1. Nalimov, V. V. (1993). V poiskah inyh smyslov. Moscow: Izdatel'skaya gruppa «Progres», 280. Available at: https://platona.net/load/knigi_po_filosofii/filosofija_poznanija/nalimov_v_poiskakh_inykh_smyslov/45-1-0-566
2. Glushkov, V. M. (1986). Kibernetika. Voprosy teorii i praktiki. Moscow: Nauka, 488. Available at: http://www.pseudology.org/science/Glushkov_Kibernetika._Voprosue_teorii_i_practiki.pdf

3. Zakrevsky, A. D. (1960). Visual-matrix method for minimization of boolean functions. Avtomatika i Telemekhanika, 21 (3), 369–373. Available at: http://www.mathnet.ru/links/fcef8cd452ff8b3804279fce2157d772/at12511.pdf
4. Plehl', O.; Yurasov, A. N. (Ed.) (1959). Elektromehanicheskaya kommutatsiya i kommutatsionnye apparaty. (Kontaktnye shemy i apparaty). Vvedenie v teoriyu i raschet. Moscow; Leningrad: Gosenergoizdat, 288. Available at: https://catalogue.nure.ua/document=101624
5. Svoboda, A. (1956). Grafico-mechanicke pomucky uzivanepfi analise a synthese kontaktovych obvodu. Stroje Zpracov. inform., 4.
6. Karnaugh, M. (1953). The map method for synthesis of combinational logic circuits. Transactions of the American Institute of Electrical Engineers, 72, 593–598.
7. Donets, S. (2015). Sources of implicit informativity of image. Filolohichni nauky, 21, 107–112. Available at: http://dspace.pnpu.edu.ua/handle/123456789/6095
8. Riznyk, V., Solomko, M. (2017). Minimization of Boolean functions by combinatorial method. Technology Audit and Production Reserves, 4 (2 (36)), 49–64. doi: https://doi.org/10.15587/2312-8372.2017.108532
9. Riznyk, V., Solomko, M. (2017). Application of super-sticking algebraic operation of variables for Boolean functions minimization by combinatorial method. Technology Audit and Production Reserves, 6 (2 (38)), 60–76. doi: https://doi.org/10.15587/2312-8372.2017.118336
10. Riznyk, V., Solomko, M. (2018). Research of 5-bit boolean functions minimization protocols by combinatorial method. Technology Audit and Production Reserves, 4 (2 (42)), 41–52. doi: https://doi.org/10.15587/2312-8372.2018.140351
11. Kondratenko, N. R. (2010). Kompiuternyi praktykum z matematychnoi lohiky. Vinnytsia: VNTU, 117. Available at: https://www.twirpx.com/file/993689/
12. Riznyk, V., Solomko, M., Tadeyev, P., Nazaruk, V., Zubyk, L., Voloshyn, V. (2020). The algorithm for minimizing Boolean functions using a method of the optimal combination of the sequence of figurative transformations. Eastern-European Journal of Enterprise Technologies, 3 (4 (105)), 43–60. doi: https://doi.org/10.15587/1729-4061.2020.206308
13. Başçiftçi, F., Akar, H. (2020). Smart minterm ordering and accumulation approach for insignificant function minimization. Ain Shams Engineering Journal. doi: https://doi.org/10.1016/j.asej.2020.04.003
14. Bernasconi, A., Ciriani, V., Villa, T. (2020). Exploiting Symmetrization and D-reducibility for Approximate Logic Synthesis. IEEE Transactions on Computers, 1–1. doi: https://doi.org/10.1109/tc.2020.3043476
15. Young, M. H., Muroga, S. (1985). Minimal covering problem and PLA minimization. International Journal of Computer & Information Sciences, 14 (6), 337–364. doi: https://doi.org/10.1007/bf00991179
16. Huang, J. (2014). Programing implementation of the Quine-McCluskey method for minimization of Boolean expression. arXiv.org. Available at: https://arxiv.org/ftp/arxiv/papers/1410/1410.1059.pdf
17. El-Bakry, H. M., Mastorakis, N. (2009). A fast computerized method for automatic simplification of boolean functions. Proceedings of the 9th WSEAS International Conference on SYSTEMS THEORY AND SCIENTIFIC COMPUTATION (ISTASC '09), 99–107. Available at: https://www.researchgate.net/profile/Hazem_El-Bakry/publication/228877182_A_fast_computerized_method_for_automatic_simplification_of_boolean_functions/links/553fa8230cf29680de9bf997/A-fast-computerized-method-for-automatic-simplification-of-boolean-functions.pdf
18. Duşa, A., Thiem, A. (2015). Enhancing the Minimization of Boolean and Multivalue Output Functions WitheQMC. The Journal of Mathematical Sociology, 39 (2), 92–108. doi: https://doi.org/10.1080/0022250x.2014.897949
19. Dusa, A. (2019). Consistency Cubes: a fast, efficient method for exact Boolean minimization. The R Journal, 10 (2), 357. doi: https://doi.org/10.32614/rj-2018-080
20. Rudell, R. L. (1989). Logic synthesis for VLSI design. Electronics Research Laboratory. Available at: http://www.cs.columbia.edu/~cs6861/handouts/rudell-PhD-thesis.pdf
21. Senchukov, V., Denysova, T. (2020). v-minimization of Boolean functions by a distance matrix and reduction to the problem of mathematical programming. Open Information and Computer Integrated Technologies, 88, 123–133. doi: https://doi.org/10.32620/oikit.2020.88.10
22. Rytsar, B. Ye. (2013). Minimization of logic functions system by konjuncterms parallel splitting method. Visnyk Natsionalnoho universytetu «Lvivska politekhnika». Radioelektronika ta telekomunikatsiyi, 766, 18–27. Available at: http://nbuv.gov.ua/UJRN/VNULPPT_2013_766_6
23. Pospelov, D. A. (1974). Logicheskie metody analiza i sinteza skhem. Moscow: «Energiya», 368. Available at: http://urss.ru/cgi-bin/db.pl?lang=Ru&blang=ru&page=Book&id=25326.
24. Shestakov, V. I. (Ed.) (1954). Sintez elektronnyh vychislitel'nyh i upravlyayushchih shem. Moscow, 358.
25. Metody minimizatsii funktsiy algebry logiki. Material iz Natsional'noy biblioteki im. N. E. Baumana. Available at: https://ru.bmstu.wiki/Методы_минимизации_функций_алгебры_логики
26. Logic Friday 1.1.4. Available at: https://www.softpedia.com/get/Others/Home-Education/Logic-Friday.shtml
27. Minimizator evristicheskoy logiki espresso. Available at: https://ru.qaz.wiki/wiki/Espresso_heuristic_logic_minimizer
28. Boolean functions' minimisation software based on the Quine-McCluskey method. Available at: http://www.seattlerobotics.org/encoder/200106/qmccmin.htm
29. JQM Java Quine McCluskey. Available at: https://sourceforge.net/projects/jqm-java-quine-mccluskey/
30. Kumar, V. D. A., Amuthan, S. G. (2016). Static structure simplification of boolean function for «n» variables – a novel approach. ICTACT Journal on Microelectronics, 1 (4), 160–167. doi: https://doi.org/10.21917/ijme.2016.0024

31. Web service Python. Available at: https://trinket.io/python/fbbf7518b8
32. Riznyk, V., Solomko, M. (2018). Minimization of conjunctive normal forms of boolean functions by combinatorial method. Technology Audit and Production Reserves, 5 (2 (43)), 42–55. doi: https://doi.org/10.15587/2312-8372.2018.146312
33. Solomko, M., Khomiuk, N., Ivashchuk, Y., Nazaruk, V., Reinska, V., Zubyk, L., Popova, A. (2020). Implementation of the method of image transformations for minimizing the Sheffer functions. Eastern-European Journal of Enterprise Technologies, 5 (4 (107)), 19–34. doi: https://doi.org/10.15587/1729-4061.2020.214899

UDC 623.462.01:531.55
DOI: 10.15587/1729-4061.2021.225594

CONSTRUCTING THE MODELS OF PROGRAMMED FLIGHT FOR PATH CALCULATION IN DESIGNING TACTICAL AND ANTI-AIRCRAFT GUIDED MISSILES

A. Chubarov
Postgraduate Student
Department of Designing and Construction
Oles Honchar Dnipro National University
Gagarina ave, 72, Dnipro, Ukraine, 49000
E-mail: achlp600@gmail.com

Several models of programmed flight have been constructed to perform calculations on flight path optimization in designing tactical and anti-aircraft-guided missiles. The developed models are based on the determination of interrelated programmed values of altitude and the flight path angle depending on the range which have a differential relationship. The combination of flight altitude and flight-path angle programs allows the users to simulate the steady flight of a guided missile to the predicted intercept endpoint using the methods of proportional control.

Good correspondence of the developed models to the physics of flight was shown by assessing the quality of approximation of the developed models of flight paths of anti-aircraft guided missiles obtained using other known models. The obtained approximation error was less than 5 % which indicates a good correspondence of the developed models to the physics of flight.

Compliance of the developed models of programmed flight with the intended purpose and the advantage over the most common known models were proved by optimizing the flight paths of the anti-aircraft-guided missile. In most of the considered calculation cases, the value of the objective function was improved to 2.9 %. The flight path was optimized using a genetic algorithm.

The developed models have a simple algebraic form and a small number of control parameters are presented in a ready-to-use form and do not require refinement for a concrete task. This allows them to be implemented in design practice without spending much time to speed up the calculation of optimal design variables and optimal flight paths of tactical and anti-aircraft-guided missiles

Keywords: missile, programmed flight model, flight path, optimization, optimal path, calculation

Received date 05.01.2021
Accepted date 08.02.2021
Published date 26.02.2021

1. Introduction

A surface-to-air missile (SAM) is a missile of the type used in anti-aircraft missile systems and designed to hit air targets [1].

A tactical missile (TM) is a type of ballistic missile designed to hit targets directly in the field of hostilities [2].

The TM and SAM designing practice shows that the choice of optimal reference paths in the process of determining design variables (DP) of missiles of these types is extremely important.

Besides, the problem of calculating the optimal paths is an important problem in aircraft designing.

The TM and SAM, as design objects, are united by the fact that their flight takes place in dense layers of the atmosphere in aero-ballistic paths. Purely ballistic paths for such missiles are not optimal as the path length increases when flying in such paths and, accordingly, the flight time increases as well. In contrast to ballistic paths, their rectifying results in large speed losses because of aerodynamic drag.

Elaboration of effective methods of optimization of design variables of TM and SAM while taking into account optimal flight paths is an urgent problem. A solution to this problem will significantly speed up the TM and SAM design process. To solve the problem of this type, it is necessary to apply analytical methods of setting various paths, i.e. the models of programmed flight. An optimal flight path is sought by varying the control parameters.

2. Literature review and problem statement

In the framework of classical aircraft design, initial DP are assessed on the basis of previous experience in design