mathematics

MDPI

Article

Polylinear Transformation Method for Solving Systems of Logical Equations

Dostonjon Numonjonovich Barotov [1,*] and Ruziboy Numonjonovich Barotov [2]

[1] Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, 4-th Veshnyakovsky Passage, 4, 109456 Moscow, Russia
[2] Department of Mathematical Analysis, Khujand State University, 1 Mavlonbekova, Khujand 735700, Tajikistan; ruzmet.tj@mail.ru
* Correspondence: dnbarotov@fa.ru

Abstract: In connection with applications, the solution of a system of logical equations plays an important role in computational mathematics and in many other areas. As a result, many new directions and algorithms for solving systems of logical equations are being developed. One of these directions is transformation into the real continuous domain. The real continuous domain is a richer domain to work with because it features many algorithms, which are well designed. In this study, firstly, we transformed any system of logical equations in the unit n-dimensional cube K_n into a system of polylinear–polynomial equations in a mathematically constructive way. Secondly, we proved that if we slightly modify the system of logical equations, namely, add no more than one special equation to the system, then the resulting system of logical equations and the corresponding system of polylinear–polynomial equations in K_{n+1} is equivalent. The paper proposes an algorithm and proves its correctness. Based on these results, further research plans are developed to adapt the proposed method.

Keywords: polylinear functions; algorithms; harmonic functions; Zhegalkin polynomials; logical operations; systems of Boolean algebraic equations; algebraic cryptanalysis; approximation; Boolean satisfiability problem; numerical optimization

MSC: 06E30; 03G05; 65H10; 90C09; 90C23; 90C26

check for updates

Citation: Barotov, D.N.; Barotov, R.N. Polylinear Transformation Method for Solving Systems of Logical Equations. *Mathematics* **2022**, *10*, 918. https://doi.org/10.3390/math10060918

Received: 30 January 2022
Accepted: 10 March 2022
Published: 13 March 2022

1. Introduction

For many decades in the history of digital science, logical variables have been the primary variables used in most computer operations. Currently, there are many theoretical and applied problems associated with logical variables, some of which lack satisfactory solutions, despite the importance of the field. One of the most important and interesting tasks is solving logical equations and systems of logical equations. Solving a system of logical equations has many applications, such as synthesis, output data coding, the state assignment of finite automata, the modeling and testing of digital networks, automatic test pattern generation and the determination of the initial state in circuits, timing analysis, and the generation of delayed failure tests for combinational circuits. The solution of a system of logical equations in the field of cryptography is used to analyze and break block ciphers, since they can be reduced to the problem of solving large-scale systems of logical equations [1–6]. This is because, for a specific cipher, algebraic cryptanalysis consists of two stages: the transformation of the cipher into a system of polynomial equations (usually over Boolean ring), and the solution of the resulting system of polynomial equations [7–9]. For example, one of the first striking applications of the solution of a system of Boolean algebraic equations in cryptography is the solution of a complex problem, cryptosystems on hidden field mappings (Hidden Fields Equations) in cryptography with a public key. This problem is described by a system of quadratic Boolean polynomials with 80 variables, and its solution was obtained precisely for the first time by solving a system of Boolean algebraic equations with the F5 algorithm, and later with the F4 algorithm [10–13].

Many of the applied algorithms for solving a system of logical equations or the Boolean satisfiability problem (SAT) that have been developed so far solve the problem in the Boolean domain. However, other areas have recently been developed and improved [11,12,14]. One of these directions is transformation into the real continuous domain. The essence of this direction is that the system of logical equations is transformed into a system in the real domain and the solution is sought in the real continuous domain. The real continuous domain is a richer domain to work with, as it involves many, well developed algorithms. Already in a real continuous domain, the transformed system can be reduced to a numerical optimization problem, making it possible to apply, analyze and combine such methods as the steepest descent algorithm, Newton's method and the coordinate descent algorithm [1–6,14].

More recently, Barotov Dostonjon et al. [14] proposed a new idea with their interesting formulas for solving systems of logical equations. The essence of their proposed method is that systems of logical equations are transformed into systems of harmonic-polynomial equations in a unit n-dimensional cube K_n with the usual operations of the addition and multiplication of numbers, and the transformed system in K_n is solved by an optimization method.

Since the proposed idea is new and under development, some of its open problems have not been solved [14]. Firstly, the corresponding system of harmonic-polynomial equations was found and the equivalence of the systems was established only when each polynomial of the system of logical equations consisted of pairwise coprime monomials. It was not clear whether there is a corresponding system of harmonic-polynomial equations in K_n for an arbitrary system of logical equations. Secondly, the equivalence of systems has not been established for any class of the system of logical equations when the solution of the system of logical equations is not unique. In our opinion, these open problems are extremely interesting and there is a demand for their solution both from the point of view of mathematics and from the point of view of the applicability and constructive improvement of the proposed method.

In this paper, we improve the proposed method from the mathematical point of view and from the point of view of algorithmization. First, we propose a constructive proof, in which for any system of logical equations, there exists in K_n a unique corresponding system of polylinear-polynomial equations. Second, we constructively prove that if we slightly modify the system of logical equations, that is, if we add no more than one special fictitious equation to the system, then for the modified system of logical equations there exists in K_{n+1} the corresponding unique equivalent system of polylinear-polynomial equations.

2. Methods and Formulas for the Transformation of Any Zhegalkin Polynomial to the Corresponding K_n Unique Non-Negative Polylinear Function

First, we define the necessary notation and formulas for further use.

Let $K_n = \{(x_1, x_2, \ldots, x_n) : 0 \le x_1, x_2, \ldots, x_n \le 1\}$ be an n-dimensional unit cube and let $B_n = \{(b_1, b_2, \ldots, b_n) : b_1, b_2, \ldots, b_n \in \{0, 1\}\}$ be the vertices of K_n.

Let $\oplus$ be the logical operation *xor* (addition by mod 2), i.e., $xor(y_1, y_2, \ldots, y_n) = y_1 \oplus y_2 \oplus \ldots \oplus y_n,\ y_i \in \{0,1\},\ \forall i \in \{1,2,\ldots,n\}$.

Let $\otimes$ be the logical operation *and* (logical multiplication), i.e., *and* $(y_1, y_2, \ldots, y_n) = y_1 \otimes y_2 \otimes \ldots \otimes y_n,\ y_i \in \{0,1\},\ \forall i \in \{1,2,\ldots,n\}$.

Let $xor_{app}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2}(1-2x_1)(1-2x_2)(1-2x_3)\ldots(1-2x_n), (x_1, x_2, \ldots, x_n) \in K_n$.

Let $and_{app}(x_1, x_2, \ldots, x_n) = x_1 \cdot x_2 \cdot \ldots \cdot x_n,\ (x_1, x_2, \ldots, x_n) \in K_n$.

In [14], the main properties of polynomials are presented and proved $xor_{app}(x_1, x_2, \ldots, x_n)$ and $and_{app}(x_1, x_2, \ldots, x_n)$.

Definition 1. *A function $f(x_1, x_2, x_3, \ldots, x_n)$ is called a polylinear function in K_n if $\frac{\partial^2}{\partial x_k^2} f(x_1, x_2, \ldots, x_n) = 0$, $\forall k \in \{1, \ldots, n\}$.*

According to Definition 1, any harmonic function in each of its variables [14] is a polylinear function, and vice versa.

Lemma 1. *For any Zhegalkin polynomial $p(x_1, x_2, x_3, \ldots, x_n) = \bigoplus_{(a_1,a_2,\ldots,a_n)\in B_n} g(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \ldots \otimes x_n^{a_n}$, there exists a non-negative, polylinear function $f(x_1, x_2, x_3, \ldots, x_n)$ on K_n such that $p(x_1, x_2, x_3, \ldots, x_n) = f(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$ and it is unique.*

Proof of Lemma 1. If $f(x_1, x_2, x_3, \ldots, x_n)$ is a polylinear function, then it has the following form

$$\begin{aligned} f(x_1, x_2, x_3, \ldots, x_n) &= c_{(0,0,0,\ldots,0)} + c_{(1,0,0,\ldots,0)} x_1 + c_{(0,1,0,\ldots,0)} x_2 + \ldots + c_{(0,0,0,\ldots,1)} x_n \\ &+ c_{(1,1,0,\ldots,0)} x_1 x_2 + c_{(1,0,1,\ldots,0)} x_1 x_3 + \ldots + c_{(0,0,\ldots,1,1)} x_{n-1} x_n + \ldots + c_{(1,1,1,\ldots,1)} x_1 x_2 \cdot \ldots \cdot x_n \\ &= \sum_{(a_1,a_2,\ldots,a_n)\in B_n} c_{(a_1,a_2,\ldots,a_n)} \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}. \end{aligned}$$

The value of the coefficients $c_{(a_1,a_2,\ldots,a_n)}$ is determined by the condition $p(x_1, x_2, x_3, \ldots, x_n) = f(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$.

$$\begin{cases} p(0,0,0,\ldots,0) = c_{(0,0,0,\ldots,0)} + 0 \\ p(1,0,0,\ldots,0) = c_{(0,0,0,\ldots,0)} + c_{(1,0,0,\ldots,0)} + 0 \\ p(0,1,0,\ldots,0) = c_{(0,0,0,\ldots,0)} + c_{(0,1,0,\ldots,0)} + 0 \\ p(1,1,0,\ldots,0) = c_{(0,0,0,\ldots,0)} + c_{(1,0,0,\ldots,0)} + c_{(0,1,0,\ldots,0)} + c_{(1,1,0,\ldots,0)} + 0 \\ \cdots\ \cdots\ \cdots\ \cdots \\ p(1,1,1,\ldots,1) = \sum_{(a_1,a_2,\ldots,a_n)\in B_n} c_{(a_1,a_2,\ldots,a_n)} \end{cases} \quad (1)$$

This lower-triangular system, that is, the corresponding matrix to the system, is lower-triangular, and it has the following unique solution:

$$\begin{cases} c_{(0,0,0,\ldots,0)} = p(0,0,0,\ldots,0) \\ c_{(1,0,0,\ldots,0)} = p(1,0,0,\ldots,0) - p(0,0,0,\ldots,0) \\ c_{(0,1,0,\ldots,0)} = p(0,1,0,\ldots,0) - p(0,0,0,\ldots,0) \\ c_{(1,1,0,\ldots,0)} = p(1,1,0,\ldots,0) - p(0,1,0,\ldots,0) - p(1,0,0,\ldots,0) + p(0,0,0,\ldots,0) \\ \cdots\ \cdots\ \cdots\ \cdots \\ c_{(1,1,1,\ldots,1)} = \sum_{(a_1,a_2,\ldots,a_n)\in B_n} (-1)^{n-(a_1+a_2+\cdots+a_n)} p(a_1, a_2, \ldots, a_n) \end{cases}$$

thus, we have proven that for any Zhegalkin polynomial $p(x_1, x_2, x_3, \ldots, x_n) = \bigoplus_{(a_1,a_2,\ldots,a_n)\in B_n} g(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \ldots \otimes x_n^{a_n}$, there is $f(x_1, x_2, x_3, \ldots, x_n)$ polylinear function such that $p(x_1, x_2, x_3, \ldots, x_n) = f(x_1, x_2, x_3, \ldots, x_n)$ for $(x_1, x_2, x_3, \ldots, x_n) \in B_n$ and it is unique, since the system (1) has a unique solution. Now, it remains to be proven that $f(x_1, x_2, x_3, \ldots, x_n)$ in K_n is non-negative. Indeed, since $f(x_1, x_2, x_3, \ldots, x_n)$ polylinear function and $p(x_1, x_2, x_3, \ldots, x_n) = f(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$, the following is true according to the maximum principle [14–16]:

$$0 \le \min_{x\in B_n} f(x_1, x_2, x_3, \ldots, x_n) \le f(x_1, x_2, x_3, \ldots, x_n) \le \max_{x\in B_n} f(x_1, x_2, x_3, \ldots, x_n) \le 1,$$

which was to be proven. □

The solution of system (1) shows that, in practice, this way of obtaining the function $f(x_1, x_2, x_3, \ldots, x_n)$ is not always optimal. This is because, in order to calculate the value of all coefficients, the values of the polynomial $p(x_1, x_2, x_3, \ldots, x_n)$ on all sets of values of the arguments $x_1, x_2, x_3, \ldots, x_n$ must be determined and then be summed in a specific way. As a result, we propose another algorithm for obtaining the function $f(x_1, x_2, x_3, \ldots, x_n)$.

The authors of [14] show that by replacing the functions $xor(x_1, x_2, \ldots, x_n)$ and $and(x_1, x_2, \ldots, x_n)$ with functions $xor_{app}(x_1, x_2, \ldots, x_n)$ and $and_{app}(x_1, x_2, \ldots, x_n)$ with any Zhegalkin polynomial $p(x_1, x_2, x_3, \ldots, x_n) = \bigoplus_{(a_1,a_2,\ldots,a_n)\in B_n} g(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \ldots \otimes x_n^{a_n}$, we can obtain the corresponding function

$$f_{old}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2} \cdot \prod_{(a_1,a_2,\ldots,a_n)\in B_n} (1 - 2 \cdot g(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}).$$

In this work, we denoted the last function as $f_{old}(x_1, x_2, \ldots, x_n)$ to emphasize that this has already been obtained in [14]. In the current work, we implement a new modified function.

From the proven properties of the formulas $xor_{app}(x_1, x_2, \ldots, x_n)$ and $and_{app}(x_1, x_2, \ldots, x_n)$ it follows [14] that $p(x_1, x_2, x_3, \ldots, x_n) = f_{old}(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$ for all Zhegalkin polynomials $p(x_1, x_2, x_3, \ldots, x_n)$. The authors of [14] prove that if the monomials $p(x_1, x_2, x_3, \ldots, x_n)$ are pairwise coprime, then the function $f_{old}(x_1, x_2, \ldots, x_n)$ is a polylinear function and $p(x_1, x_2, x_3, \ldots, x_n) = f_{old}(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$.

If the monomials are $p(x_1, x_2, x_3, \ldots, x_n)$ and not pairwise coprime, then we modify the function $f_{old}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2} \cdot \prod_{(a_1,a_2,\ldots,a_n)\in B_n} (1 - 2 \cdot g(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n})$ and show an algorithm for obtaining a unique, non-negative, polylinear function $f_{new}(x_1, x_2, x_3, \ldots, x_n)$ on K_n such that $p(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$.

On this occasion, the first basic idea is that in the expansion of the polynomial

$$f_{old}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2} \cdot \prod_{(a_1,a_2,\ldots,a_n)\in B_n} (1 - 2 \cdot g(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n})$$

if all degrees x_i^k where $k = 2, 3, 4, \ldots$ is replaced by x_i, then the resulting polynomial is $f_{new}(x_1, x_2, x_3, \ldots, x_n)$ or, in other words:

$$f_{new}(x_1, x_2, x_3, \ldots, x_n) = f(x_1, x_2, \ldots, x_n) \ / < x_1^2 - x_1, x_2^2 - x_2, \ldots, x_n^2 - x_n >.$$

The second main idea is that, without expanding the polynomial mathematically, $f_{old}(x_1, x_2, \ldots, x_n)$ replaces all degrees x_i^k where $k = 2, 3, 4, \ldots$ on x_i and produces $f_{new}(x_1, x_2, x_3, \ldots, x_n)$, since in the expanded form of the polynomial, the $f_{old}(x_1, x_2, \ldots, x_n)$ number of terms is probably large. If we expand the polynomial $f_{old}(x_1, x_2, \ldots, x_n)$ and group by degrees x_i, then it has the following form:

$$f_{old}(x_1, x_2, \ldots, x_n) = a_m x_i^m + a_{m-1} x_i^{m-1} + \ldots + a_2 x_i^2 + a_1 x_i + a_0,$$

where the coefficients are $a_j = a_j(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, with some polynomials depending on $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ and not depending on x_i. If all degrees are x_i^k where $k = 2, 3, 4, \ldots, m$ replace with x_i, then the resulting polynomial appears as $a_m x_i + a_{m-1} x_i + \ldots + a_2 x_i + a_1 x_i + a_0$, and this polynomial can still be implemented as follows:

$$\begin{aligned} a_m x_i + a_{m-1} x_i + \ldots + a_2 x_i + a_1 x_i + a_0 &= (a_m + a_{m-1} + \ldots + a_2 + a_1 + a_0) x_i - a_0 x_i + a_0 \\ &= f_{old}(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) x_i - f_{old}(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)(x_i - 1) \\ &= f_{old}(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) x_i + f_{old}(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)(1 - x_i) \end{aligned}$$

The following algorithm is based on this.

Let $X_{degree} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ be the set of all variables such that each variable x_{i_s} is included in at least two monomials of the polynomial $p(x_1, x_2, x_3, \ldots, x_n)$. If another variable occurs in at most one monomial of polynomial $p(x_1, x_2, x_3, \ldots, x_n)$, then the polynomial $f_{old}(x_1, x_2, \ldots, x_n)$ is, at most, first-degree in that variable. We are ready to write an algorithm for obtaining $f_{new}(x_1, x_2, x_3, \ldots, x_n)$ from the polynomial $f_{old}(x_1, x_2, x_3, \ldots, x_n)$ (Algorithm 1).

Algorithm 1: Obtaining $f_{new}(x_1, x_2, x_3, \ldots, x_n)$ from the polynomial $f_{old}(x_1, x_2, x_3, \ldots, x_n)$

```
Input: polynomial $f_{old}(x_1, x_2, x_3, \ldots, x_n)$ and set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$.
Output: polynomial $f_{new}(x_1, x_2, x_3, \ldots, x_n)$
$X_{degree} := \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$
$f_0(x_1, x_2, x_3, \ldots, x_n) := f_{old}(x_1, x_2, x_3, \ldots, x_n)$
for $s$ from 1 to $k$
do
    $f_s(x_1, x_2, x_3, \ldots, x_n) := f_{s-1}(x_1, x_2, \ldots, x_{i_s-1}, 1, x_{i_s+1}, \ldots, x_n)x_{i_s} +$
    $f_{s-1}(x_1, x_2, \ldots, x_{i_s-1}, 0, x_{i_s+1}, \ldots, x_n)(1 - x_{i_s})$
end for
$f_{new}(x_1, x_2, x_3, \ldots, x_n) := f_k(x_1, x_2, x_3, \ldots, x_n)$
return $f_{new}(x_1, x_2, x_3, \ldots, x_n)$
```

Next, we prove that Algorithm 1 is correct:

(i) $p(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$.
(ii) $f_{new}(x_1, x_2, x_3, \ldots, x_n)$ is a polylinear function.
(iii) If $(x_1, x_2, x_3, \ldots, x_n) \in K_n$, then $0 \le f_{new}(x_1, x_2, x_3, \ldots, x_n) \le 1$.
(iv) $f_{new}(x_1, x_2, x_3, \ldots, x_n)$ is unique and polylinear function such that $p(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$.

Proof of Algorithm 1. (i) In this article, we have already established that $p(x_1, x_2, x_3, \ldots, x_n) = f_{old}(x_1, x_2, x_3, \ldots, x_n)$ for $(x_1, x_2, x_3, \ldots, x_n) \in B_n$. Furthermore, it is clear that $0^k = 0$ and $1^k = 1$ and, therefore, $f_{old}(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, x_3, \ldots, x_n)$ for $(x_1, x_2, x_3, \ldots, x_n) \in B_n \Rightarrow p(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, x_3, \ldots, x_n)$ as $(x_1, x_2, x_3, \ldots, x_n) \in B_n$.

(ii) To prove this point of the assertion, it suffices to show that $\frac{\partial^2}{\partial x_k^2}(f_{new}(x_1, x_2, \ldots, x_n)) = 0\ \forall k \in \{1, 2, \ldots, n\}$. If $x_k \notin \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$, then $\frac{\partial^2}{\partial x_k^2}(f_{old}(x_1, x_2, \ldots, x_n)) = 0 \Rightarrow \frac{\partial^2}{\partial x_k^2}(f_{new}(x_1, x_2, \ldots, x_n)) = 0$, and if $x_k \in \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$, then $\frac{\partial^2}{\partial x_k^2}(f_{new}(x_1, x_2, \ldots, x_n)) = \frac{\partial^2}{\partial x_k^2}(f_k(x_1, x_2, x_3, \ldots, x_n)) = \frac{\partial^2}{\partial x_k^2}(a_m x_k + a_{m-1}x_k + \ldots + a_2 x_k + a_1 x_k + a_0) = 0$.

(iii) From the properties $(i) - (ii)$ and from the maximum principle, the following inequality follows [14,17–19]:

$$0 \le \min_{x \in B_n} f_{new}(x_1, x_2, \ldots, x_n) \le f_{new}(x_1, x_2, \ldots, x_n) \le \max_{x \in B_n} f_{new}(x_1, x_2, \ldots, x_n) \le 1.$$

(iv) Prove by contradiction, let there be another non-negative polylinear function $h(x_1, x_2, \ldots, x_n)$ $(h(x_1, x_2, \ldots, x_n) \ne f_{new}(x_1, x_2, \ldots, x_n))$ such that $h(x_1, x_2, \ldots, x_n) = p(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n$. Now, let us observe the function $d(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, \ldots, x_n) - h(x_1, x_2, \ldots, x_n)$. Firstly, it is obvious that if $(x_1, x_2, x_3, \ldots, x_n) \in B_n$, then $d(x_1, x_2, x_3, \ldots, x_n) = f_{new}(x_1, x_2, \ldots, x_n) - h(x_1, x_2, \ldots, x_n) = p(x_1, x_2, x_3, \ldots, x_n) - p(x_1, x_2, x_3, \ldots, x_n) = 0$. Secondly, the function $d(x_1, x_2, x_3, \ldots, x_n)$ is polylinear, because $\frac{\partial^2}{\partial x_k^2} d(x_1, x_2, \ldots, x_n) = \frac{\partial^2}{\partial x_k^2} f(x_1, x_2, \ldots, x_n) - \frac{\partial^2}{\partial x_k^2} h(x_1, x_2, \ldots, x_n) = 0 - 0 = 0\ \forall k \in \{1, \ldots, n\}$. It now follows from the last argument and from the maximum principle [14,20–22] that

$$0 = \min_{x \in B_n} d(x_1, x_2, x_3, \ldots, x_n) \le d(x_1, x_2, x_3, \ldots, x_n) \le \max_{x \in B_n} d(x_1, x_2, x_3, \ldots, x_n) = 0 \Rightarrow$$

$d(x_1, x_2, x_3, \ldots, x_n) \equiv 0 \Rightarrow h(x_1, x_2, \ldots, x_n) \equiv f(x_1, x_2, \ldots, x_n)$ is a contradiction. □

3. Transformation of a System of Logical Equations into a System of Polylinear-Polynomial Equations

After proving the correctness of Algorithm 1, we are ready to transform an arbitrary system of Boolean algebraic equations into a system of polylinear-polynomial equations in a unit n-dimensional cube K_n with the usual operations of the addition and multiplication of numbers.

Consider the following arbitrary system of Boolean algebraic equations:

$$\begin{cases} p_1(x_1, x_2, \ldots, x_n) = \bigoplus\limits_{(a_1,a_2,\ldots,a_n)\in B_n} g_1(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \ldots \otimes x_n^{a_n} = 0 \\ p_2(x_1, x_2, \ldots, x_n) = \bigoplus\limits_{(a_1,a_2,\ldots,a_n)\in B_n} g_2(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \ldots \otimes x_n^{a_n} = 0 \\ \ldots \\ p_m(x_1, x_2, \ldots, x_n) = \bigoplus\limits_{(a_1,a_2,\ldots,a_n)\in B_n} g_3(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \ldots \otimes x_n^{a_n} = 0 \end{cases} \quad (2)$$

where is $p_i(x_1, x_2, \ldots, x_n)$ a Zhegalkin polynomial, $i \in \{1, 2, \ldots, m\}$.

Replacing functions $xor(x_1, x_2, \ldots, x_n)$ and $and(x_1, x_2, \ldots, x_n)$ in system (2), we obtain an intermediate transformationed system:

$$\begin{cases} f_{old_1}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2} \cdot \prod\limits_{(a_1,a_2,\ldots,a_n)\in B_n} \left(1 - 2 \cdot g_1(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}\right) = 0 \\ f_{old_2}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2} \cdot \prod\limits_{(a_1,a_2,\ldots,a_n)\in B_n} \left(1 - 2 \cdot g_2(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}\right) = 0 \\ f_{old_m}(x_1, x_2, \ldots, x_n) = \frac{1}{2} - \frac{1}{2} \cdot \prod\limits_{(a_1,a_2,\ldots,a_n)\in B_n} \left(1 - 2 \cdot g_m(a_1, a_2, \ldots, a_n) \cdot x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}\right) = 0 \end{cases} \quad (3)$$

Modifying each polynomial of the $f_{old_i}(x_1, x_2, \ldots, x_n)$ intermediate system (3) according to Algorithm 1, we obtain the corresponding system of polylinear-polynomial equations:

$$\begin{cases} f_{new_1}(x_1, x_2, \ldots, x_n) = 0 \\ f_{new_2}(x_1, x_2, \ldots, x_n) = 0 \\ f_{new_3}(x_1, x_2, \ldots, x_n) = 0 \\ \ldots \\ f_{new_m}(x_1, x_2, \ldots, x_n) = 0 \end{cases} \quad (4)$$

For system (4), define the following objective function in K_n:

$$tf(x_1, x_2, \ldots, x_n) := \sum_{i=1}^{m} f_{new_i}(x_1, x_2, \ldots, x_n)$$

Below, we formulate and check the main properties of the objective function $tf(x_1, x_2, \ldots, x_n)$.

Proposition 1. *Let* $tf(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{m} f_i(x_1, x_2, \ldots, x_n)$ *and* $(x_1, x_2, \ldots, x_n) \in K_n$*, then:*

(i) *In K_n, the objective function $tf(x_1, x_2, \ldots, x_n)$ is on-negative.*
(ii) *The objective function $tf(x_1, x_2, \ldots, x_n)$ does not have a local extreme inside edges and faces of the n—dimensional cube K_n.*
(iii) *Takes its extreme at the vertices of the n—dimensional cube K_n, that is, on B_n.*
(iv) *If $(b_1, b_2, \ldots, b_n) \in B_n$ is a solution to the system (2), then $(b_1, b_2, \ldots, b_n)$ is a solution to the system (4) and $tf(b_1, b_2, \ldots, b_n) = 0$.*
(v) *$(s_1^*, s_2^*, .., s_n^*) \in K_n$ is the solution to the system (4) $\Leftrightarrow$ $tf(s_1^*, s_2^*, .., s_n^*) = 0$.*

Proof of Proposition 1. (i) It follows from Algorithm 1 that if $(x_1, x_2, \ldots, x_n) \in K_n$, then $0 \le f_{new_i}(x_1, x_2, \ldots, x_n) \le 1 \; \forall i \in \{1, 2, \ldots, m\} \Rightarrow$ if $(x_1, x_2, \ldots, x_n) \in K_n$, then $0 = \sum_{i=1}^{m} 0 \le \sum_{i=1}^{m} f_{new_i}(x_1, x_2, \ldots, x_n) = tf(x_1, x_2, \ldots, x_n) \le \sum_{i=1}^{m} 1 = m$.

To prove parts (ii) and (iii), it suffices to show that $\frac{\partial^2}{\partial x_k^2} tf(x_1, x_2, \ldots, x_n) = 0\ \forall k \in \{1, 2, \ldots, n\}$ is really $\frac{\partial^2}{\partial x_k^2} tf(x_1, x_2, \ldots, x_n) = \frac{\partial^2}{\partial x_k^2} (\sum_{i=1}^{m} f_{new_i}(x_1, x_2, \ldots, x_n)) = \sum_{i=1}^{m} \frac{\partial^2}{\partial x_k^2} f_{new_i}$ $(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{m} 0 = 0$.

(iv) If $(b_1, b_2, \ldots, b_n) \in B_n$ is a solution to system (2), then it $p_i(b_1, b_2, \ldots, b_n) = 0\ \forall i \in \{1, 2, \ldots, m\}$ also follows from Algorithm 1 that $f_{new_i}(b_1, b_2, \ldots, b_n) = 0\ \forall i \in \{1, 2, \ldots, m\}$. This means that $(b_1, b_2, \ldots, b_n) \in B_n$ is a solution to system (4) and $tf(b_1, b_2, \ldots, b_n) = \sum_{i=1}^{m} f_{new_i}(b_1, b_2, \ldots, b_n) = \sum_{i=1}^{m} 0 = 0$.

(v) If $(s_1^*, s_2^*, .., s_n^*) \in K_n$ is a solution to system (4), then $f_{new_i}(s_1^*, s_2^*, .., s_n^*) = 0$ $\forall i \in \{1, 2, \ldots, m\} \Rightarrow\ tf(s_1^*, s_2^*, .., s_n^*) = 0$ and if $tf(s_1^*, s_2^*, .., s_n^*) = 0$ for $(s_1^*, s_2^*, .., s_n^*) \in K_n$, then it follows from Algorithm 1 that $f_{new_i}(s_1^*, s_2^*, .., s_n^*) = 0\ \forall i \in \{1, 2, \ldots, m\}$. This means that $(s_1^*, s_2^*, .., s_n^*)$ is a solution to system (4). □

Theorem 1. *If system*(2) *has, at most, one solution, then in* K_n *system* (4) *and system* (2) *are equivalent.*

Proof of Theorem 1. *Case 1*. Let system (2) have no solution. We prove that system (4) in K_n does not have a solution either. Proof by contradiction: let system (4) in K_n have the following solution $(c_1, c_2, \ldots, c_n)$, because $(c_1, c_2, \ldots, c_n) \in K_n$. Therefore, it follows from Proposition 1 that $tf(c_1, c_2, \ldots, c_n) = 0$, but, on the other hand, $tf(x_1, x_2, \ldots, x_n)$ is a polylinear function and the following inequality follows from Algorithm 1:

$0 < \underset{x \in B_n}{min}\ tf(x_1, x_2, \ldots, x_n) \leq tf(x_1, x_2, \ldots, x_n) \leq \underset{x \in B_n}{max}\ tf(x_1, x_2, \ldots, x_n) \leq m$, $\forall (x_1, x_2, \ldots, x_n) \in K_n$. Since system (2) has no solution, the first inequality on the left is strict. If in the last inequality we substitute $(c_1, c_2, \ldots, c_n) \in K_n$ for $(x_1, x_2, \ldots, x_n)$, then we obtain the following inequality: $0 < \underset{x \in B_n}{min}\ tf(x_1, x_2, \ldots, x_n) \leq tf(c_1, c_2, \ldots, c_n) = 0 \leq \underset{x \in B_n}{max}\ tf(x_1, x_2, \ldots, x_n) \leq m$ contradiction, which was to be proven.

Case 2. Let system (2) have a unique solution $(x_1^*, x_2^*, \ldots, x_n^*)$. In this case, we prove that system (4) in K_n has a unique solution $(x_1^*, x_2^*, \ldots, x_n^*)$. According to Algorithm 1 $p_i(x_1, x_2, x_3, \ldots, x_n) = f_{new_i}(x_1, x_2, x_3, \ldots, x_n)$ at $(x_1, x_2, x_3, \ldots, x_n) \in B_n\ \forall i \in \{1, 2, \ldots, m\}$. In particular, it follows that $p_i(x_1^*, x_2^*, \ldots, x_n^*) = f_{new_i}(x_1^*, x_2^*, \ldots, x_n^*) = 0\ \forall i \in \{1, 2, \ldots, m\}$, which means that $(x_1^*, x_2^*, \ldots, x_n^*)$ is a solution to system (4). Now, we prove that in K_n it is the only solution of the system (4). Let us prove by contradiction. Let $(c_1, c_2, \ldots, c_n) \in K_n$ be another solution of system (4). According to Proposition 1 $(c_1, c_2, \ldots, c_n) \in K_n$ is the solution to system (4) $\Leftrightarrow tf(c_1, c_2, \ldots, c_n) = 0$. Since the system (7) has a unique solution and the objective function $tf(x) = tf(x_1, x_2, \ldots, x_n)$ is polylinear, it follows that $(c_1, c_2, \ldots, c_n) \in B_n$, which means that $f_{new_i}(c_1, c_2, \ldots, c_n) = p_i(c_1, c_2, \ldots, c_n) = 0$ $\forall i \in \{1, 2, \ldots, m\} \Rightarrow (c_1, c_2, \ldots, c_n)$ is a solution to the system (2), since, according to the condition *Case 2* of the theorem, system (2) has a unique solution $(x_1^*, x_2^*, \ldots, x_n^*) \Rightarrow$ $(x_1^*, x_2^*, \ldots, x_n^*) = (c_1, c_2, \ldots, c_n)$ is a contradiction, which was proven. □

If the number of solutions to system (2) is at least two, then, in general, the equivalence of systems (2) and (4) is violated. For clarity, consider a few counterexamples:

$$a) \begin{cases} x_1 \otimes x_2 \oplus x_1 = 0 \\ x_1 \otimes x_2 \oplus 1 = 1 \end{cases}, \quad b) \begin{cases} x_1 \otimes x_2 \oplus x_3 = 0 \\ x_1 \oplus x_3 \oplus 1 = 1 \end{cases}, \quad c) \begin{cases} x_1 \otimes x_2 \oplus x_1 = 0 \\ x_1 \otimes x_3 \oplus x_1 = 0 \\ x_1 \otimes x_2 \oplus 1 = 1 \end{cases}, \ldots$$

Indeed, we can transform these systems in K_n in systems of polylinear-polynomial equations with the usual operations of the addition and multiplication of numbers:

$$a') \begin{cases} x_1 - x_1x_2 = 0 \\ x_1x_2 = 0 \end{cases}, \quad b') \begin{cases} x_1x_2 + x_3 - 2x_1x_2x_3 = 0 \\ x_1 + x_3 - 2x_1x_3 = 0 \end{cases}, \quad c') \begin{cases} x_1 - x_1x_2 = 0 \\ x_1 - x_1x_3 = 0 \\ x_1x_2 = 0 \end{cases}, \ldots$$

Solving these systems in K_n, compare the corresponding solutions:

- the system (a) has solutions $(0,0) \cup (0,1)$, and the system (a') has solutions $(0,x_2)$, where is x_2 any number from the segment $[0,1]$.
- the system (b) has solutions $(0,0,0) \cup (0,1,0) \cup (1,1,1)$, and the system (b') has solutions $(0,x_2,0) \cup (1,1,1)$, where is x_2 any number from the segment $[0,1]$.
- the system (c) has solutions $(0,0,0) \cup (0,0,1) \cup (0,1,0) \cup (0,1,1)$, and the system (c') has solutions $(0,x_2,x_3)$, where x_2 and x_3 are any numbers from the segment $[0,1]$.

4. Method for Obtaining Equivalent Systems in K_{n+1}

In this section, we slightly modify system (2); namely, we add one fictitious equation to the system and prove the equivalence of new systems.

Let $\{x_{j_1}, x_{j_2}, \dots, x_{j_l}\}$ be the set of all the variables, such that each variable x_{j_s} $s = 1,2,\dots,l$ occurs only in monomials of at least the second degree. In this case, the modified system of Boolean algebraic equations has the following form:

$$\begin{cases} p_1(x_1,x_2,\dots,x_n) = \bigoplus\limits_{(a_1,a_2,\dots,a_n)\in B_n} g_1(a_1,a_2,\dots,a_n)\cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \dots \otimes x_n^{a_n} = 0 \\ p_2(x_1,x_2,\dots,x_n) = \bigoplus\limits_{(a_1,a_2,\dots,a_n)\in B_n} g_2(a_1,a_2,\dots,a_n)\cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \dots \otimes x_n^{a_n} = 0 \\ p_m(x_1,x_2,\dots,x_n) = \bigoplus\limits_{(a_1,a_2,\dots,a_n)\in B_n} g_m(a_1,a_2,\dots,a_n)\cdot x_1^{a_1} \otimes x_2^{a_2} \otimes \dots \otimes x_n^{a_n} = 0 \\ x_{j_1} \oplus x_{j_2} \oplus \dots \oplus x_{j_l} \oplus x_{n+1} = 0 \end{cases} \tag{5}$$

Replacing the functions $xor(x_1,x_2,\dots,x_n)$ and $and(x_1,x_2,\dots,x_n)$ in system (5), we obtain an intermediate transformationed system:

$$\begin{cases} f_{old_1}(x_1,x_2,\dots,x_n) = \frac{1}{2} - \frac{1}{2}\cdot \prod\limits_{(a_1,a_2,\dots,a_n)\in B_n} (1 - 2\cdot g_1(a_1,a_2,\dots,a_n)\cdot x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}) = 0 \\ f_{old_2}(x_1,x_2,\dots,x_n) = \frac{1}{2} - \frac{1}{2}\cdot \prod\limits_{(a_1,a_2,\dots,a_n)\in B_n} (1 - 2\cdot g_2(a_1,a_2,\dots,a_n)\cdot x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}) = 0 \\ f_{old_m}(x_1,x_2,\dots,x_n) = \frac{1}{2} - \frac{1}{2}\cdot \prod\limits_{(a_1,a_2,\dots,a_n)\in B_n} (1 - 2\cdot g_m(a_1,a_2,\dots,a_n)\cdot x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}) = 0 \\ f_{old_{m+1}}(x_1,x_2,\dots,x_n,x_{n+1}) = \frac{1}{2} - \frac{1}{2}\cdot(1-2x_{j_1})(1-2x_{j_2})\cdot\ldots\cdot(1-2x_{j_l})(1-2x_{n+1}) = 0 \end{cases} \tag{6}$$

Modifying each polynomial of the $f_{old_i}(x_1,x_2,\dots,x_n)$ intermediate system (6) according to Algorithm 1, we obtain the corresponding system of polylinear-polynomial equations:

$$\begin{cases} f_{new_1}(x_1,x_2,\dots,x_n) = 0 \\ f_{new_2}(x_1,x_2,\dots,x_n) = 0 \\ f_{new_3}(x_1,x_2,\dots,x_n) = 0 \\ \dots \\ f_{new_m}(x_1,x_2,\dots,x_n) = 0 \\ f_{new_{m+1}}(x_1,x_2,\dots,x_n,x_{n+1}) = 0 \end{cases} \tag{7}$$

For system (7), define the following objective function in K_{n+1}:

$$tf(x_1,x_2,\dots,x_n,x_{n+1}) := \sum_{i=1}^{m} f_{new_i}(x_1,x_2,\dots,x_n) + f_{new_{m+1}}(x_1,x_2,\dots,x_n,x_{n+1}).$$

Now we are ready to formulate and prove the theorem of the equivalence of systems (5) and (7).

Theorem 2. *In K_{n+1}, system* (5) *and system* (7) *are equivalent.*

Proof of Theorem 2. (i) If $(x_1^*, x_2^*, \dots, x_n^*, x_{n+1}^*) \in K_{n+1}$ is a solution to system (5), then from the Algorithm 1 and from the properties of formulas $xor_{app}(x_1,x_2,\dots,x_n)$ and $and_{app}(x_1,x_2,\dots,x_n)$ it follows [14] that $(x_1^*, x_2^*, \dots, x_n^*, x_{n+1}^*)$ is a solution to system (7). Conversely, if $(x_1^*, x_2^*, \dots, x_n^*, x_{n+1}^*) \in K_{n+1}$ is a solution to system (7), then by the form of the last equation of system (7) and from the properties of the formulas $xor_{app}(x_1, x_2, \ldots, x_n)$ and $and_{app}(x_1, x_2, \ldots, x_n)$, it follows [14] that $\left(x_1^*, x_2^*, \ldots, x_n^*, x_{n+1}^*\right) \in B_{n+1}$. Now, if $\left(x_1^*, x_2^*, \ldots, x_n^*, x_{n+1}^*\right) \in B_{n+1}$, then according to Algorithm 1 and from the properties of the formulas it $xor_{app}(x_1, x_2, \ldots, x_n)$ follows that:

$f_{new_i}\left(x_1^*, x_2^*, \ldots, x_n^*\right) = p_i\left(x_1^*, x_2^*, \ldots, x_n^*\right)\ \forall i \in \{1, 2, \ldots, m\}$ and $x_{n+1}^* = x_{j_1}^* \oplus x_{j_2}^* \oplus \ldots \oplus x_{j_l}^*$ or $\left(x_1^*, x_2^*, \ldots, x_n^*, x_{n+1}^*\right)$ is the solution of system (5).

(ii) If system (5) has no solution, then Theorem 1 implies that in K_{n+1}, system (7) also has no solution. Conversely, if system (7) does not have a solution, then system (5) also does not have a solution, since the last equation added to systems (5)–(7) allows us to say that in K_{n+1}, any solution to system (7) belongs to B_{n+1}. $\square$

5. Conclusions

In this paper, firstly, we managed to transform any system of logical equations in K_n into a system of polylinear-polynomial equations. Secondly, we proved the correctness of such a transformation, and the corresponding proof was mathematically constructive. Based on this, we proposed a transformation algorithm and proved its correctness.

In addition, we also proved that if we slightly modify the system of logical equations, namely, if we add no more than one special equation to the system, then, firstly, all the solutions of the unmodified system of logical equations are the initial n coordinates of the solutions of the modified system of logical equations, and vice versa. Secondly, the modified system of logical equations and the corresponding system of polylinear-polynomial equations in K_{n+1} are equivalent.

In terms of transformation into a real continuous domain and the application of numerical optimization methods to solving a transformed system into a real continuous domain, that is, minimizing the objective function corresponding to the transformed system, then there are many approaches to transformation into a real continuous domain. However, one of the main problems with this is that the objective function in the desired continuous domain can have many local minimums, and this greatly hinders the suitability of the method in practice. Currently, as far as we know, all numerical optimization algorithms find only a local minimum, and, in general, there is no effective method for finding the global minimum of the objective function. As far as we know, our transformation method is the most optimal in terms of reducing the number of local minimums of the objective function $tf(x_1, x_2, \ldots, x_n)$, because the minimized objective function compiled according to the system of logical equations is such that it is polylinear for any system of logical equations. Therefore, in K_n, this minimizing objective function does not have a local extremum inside, or on the edges and faces of K_n and takes a minimum value at the vertices of K_n.

We must also admit that, firstly, it is not clear to what extent these theoretical results will be suitable in practice. Secondly, if we want to solve the transformed system of polylinear-polynomial equations using numerical optimization methods, then the problem of developing, improving and finding the asymptotic complexity of the numerical optimization algorithm used for our particular polylinear objective function remains open. Therefore, based on the results of the current work, in future work, we plan to publish a comparative analysis of the most well-known optimization methods to minimize our specific polylinear function.

Author Contributions: Formal analysis, D.N.B. and R.N.B.; Methodology, D.N.B.; Writing—review and editing, R.N.B. and D.N.B.; Validation, D.N.B. and R.N.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abdel-Gawad, A.H.; Atiya, A.F.; Darwish, N.M. Solution of systems of Boolean equations via the integer domain. *Inform. Sci.* **2010**, *180*, 288–300. [CrossRef]
2. Barotov, D.N.; Muzafarov, D.Z.; Barotov, R.N. On one method for solving systems of Boolean algebraic equations. *Mod. Math. Concept Innov. Math. Educ.* **2021**, *8*, 17–23.
3. Gu, J.; Gu, Q.; Du, D. On optimizing the satisfiability (SAT) problem. *J. Comput. Sci. Technol.* **1999**, *14*, 1–17. [CrossRef]
4. Gu, J. Global optimization for satisfiability (SAT) problem. *IEEE Trans. Knowl. Data Eng.* **1994**, *6*, 361–381. [CrossRef]
5. Gu, J. *How to Solve Very Large-Scale Satisfiability (VLSS) Problems*; Technical Report UCECETR-90-002; University of Calgary: Calgary, AB, Canada, 1990.
6. Gu, J. On optimizing a search problem. In *Artificial Intelligence Methods and Applications*; Bourbakis, N.G., Ed.; World Scientific Publishers: Singapore, 1992.
7. Armknecht, F. Improving Fast Algebraic Attacks. In *International Workshop on Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 65–82.
8. Bardet, M.; Faugèrebcd, J.-C.; Salvye, B.; Spaenlehauer, P.-J. On the complexity of solving quadratic boolean systems. *J. Complex.* **2013**, *29*, 53–75. [CrossRef]
9. Courtois, N. *Fast Algebraic Attacks on Stream Ciphers with Linear Feedback*; CRYPTO 2003, Lecture Notes in Computer Science, 2729; Boneh, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 176–194.
10. Bard, G.V. *Algorithms for Solving Linear and Polynomial Systems of Equations over Finite Fields, with Applications to Cryptanalysis*; University of Maryland: College Park, MD, USA, 2007.
11. Faugere, J.C. A new efficient algorithm for computing Gröbner bases (F4). *J. Pure Appl. Algebra* **1999**, *139*, 61–88. [CrossRef]
12. Faugere, J.C. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, Lille, France, 7–10 July 2002; pp. 75–83.
13. Liu, M.; Lin, D.; Pei, D. Fast algebraic attacks and decomposition of symmetric Boolean functions. *IEEE Trans. Inf. Theory* **2011**, *57*, 4817–4821. [CrossRef]
14. Barotov, D.; Osipov, A.; Korchagin, S.; Pleshakova, E.; Muzafarov, D.; Barotov, R.; Serdechnyy, D. Transformation Method for Solving System of Boolean Algebraic Equations. *Mathematics* **2021**, *9*, 3299. [CrossRef]
15. Axler, S.; Bourdon, P.; Wade, R. *Harmonic Function Theory*, 2nd ed.; Springer: New York, NY, USA, 2001; Volume 137.
16. Kosmodem'yanskii, A.A. A converse of the mean value theorem for harmonic functions. *Russ. Math. Surv.* **1981**, *36*, 159.
17. Barotov, D.N.; Barotov, R.N. Representation of a particular solution of a differential equation with singular coefficients. *Sci. Notes Khujand State Univ. Named After Acad. B. Gafurov. Ser. Nat. Sci. Econ.* **2018**, *2*, 3–8.
18. Connolly, C.I.; Grupen, R.A. The applications of harmonic functions to robotics. *J. Robot. Syst.* **1993**, *10*, 931–946. [CrossRef]
19. Goldstein, M.; Ow, W.H. On the mean-value property of harmonic functions. *Proc. Am. Math. Soc.* **1971**, *29*, 341–344. [CrossRef]
20. Epstein, B.; Schiffer, M.M. On the mean-value property of harmonic functions. J. *D'Analyse Mathématique* **1965**, *14*, 109–111. [CrossRef]
21. Mukhsinov, A.; Boboev, E.D.; Barotov, D.N. Formula presentation the solutions of the initial-boundary problem for one a many-dimtnsional parabolic equation with singular coefficients. *Sci. Notes Khujand State Univ. Named After Acad. B. Gafurov. Ser. Nat. Sci. Econ.* **2016**, *4*, 18–25.
22. Zhang, D.; Zhang, G.; Zheng, E. The harmonic polynomial method for solving the Cauchy problem connected with the Laplace equation. *Inverse Probl.* **2013**, *29*, 065008. [CrossRef]