Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

Annales UMCS Informatica AI XIII, 1 (2013) 11–21
DOI: 10.2478/v10065-012-0043-x

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

A set of connection network synthesis based on the linear Diophantine constraints solution in area {0,1}

Marek Bolanowski[1*], Andrzej Paszkiewicz[1†]

[1] *Departament of Distributed Systems, Rzeszow University of technology, Pola 2, 35-959 Rzeszow, Poland*

Abstract – In the paper we present a method for synthesis of a set of connection networks, which fulfill the designer's assumption. In order to do that, the designed network is presented as a system of Diophantine constraints in a special form. A method to resolve those systems is proposed, which is characterized by lower computational complexity, especially in the case of rare topologies design.

1 Introduction

Efficiency of the computational systems synthesis depends on the accurate connection network choice, and possibilities of its reconfiguration during the computational system life time. The selection of the connection structure for large scale systems has to be made automatically by using algorithms, which generate the solutions in an acceptable time period. Created solutions have to be characterized by good technical and exploitation parameters which can be described by the following relation:

$$\text{opt } \Pi(X) = \min \Pi(X) = E_n K_n(X) + \Im(X) + \Upsilon(X)$$

where: $\Pi(X)$ - the system building and exploitation costs; E_n - the amortization rate; $K_n(X)$ - the realization costs; $\Im(X)$ - the exploitation costs, $\Upsilon(X)$ - the utilization costs; X - the table of the designing system parameters [**1**].

Processes connected with the synthesis and analysis interconnection network are characterized by high computational complexity as a result of: hard formalization of the designing process constraint; the necessity to use the source data represented by a wide value range or approximate meaning; multiplicity of the designing process. The

*mb@prz.edu.pl
†andrzejp@prz.edu.pl

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

high complexity of the resolving complex system synthesis task requires its decomposition which allows for the transition from one gross, high complexity designing task to the sequence of smaller designing tasks. The decomposition process can be made simultaneously at the structural level and different synthesis subtask levels [**1**].

Unfortunately, nowadays, there are no methods to resolve the task of topology set synthesis with specified parameters which are characterized by acceptable computational complexity. The goal of this work is to elaborate the effective methods for synthesis of the connection networks set, based on resolving of the system of linear Diophantine constraints in the area {0,1}.

Let us assume that the designed computational system will use the unidirectional communication channels, therefore, a directed graph will be used to represent it.

The directed graph is a pair $G = (V, E)$, for which $V = \{v_1, v_2, \ldots, v_n\}$ is the set of elements called vertices, and the set $E \subseteq V \times V$is the set of elements called arcs where $V \times V$ is the Cartesian square of set V. A graph will be called labelled (weighted) when: for the presentation of the elements from the sets $V$and/or E there is used not only the identifier, but it is necessary to use the additional variables (edge and/or vertices weight) to store some characteristics or attributes. If the weights of vertices or edges are not specified then such a graph will be called an unlabelled graph.

The adjacency matrix $X = [x_{i,j}]$ will be called the square matrix with the dimension $n \times n$ elements, where n - the number of the graph vertices, which are described as follows: $x_{i,j} = 1$ if in the graph G there is an edge connecting the vertices v_i and v_j; $x_{i,j} = 0$, if there is no edge connecting the vertices v_i and v_j [**2, 3, 4**].

The sum R_i of the elements of the matrix X in relation to rows i.e.:

$$R_i = \sum_{\alpha=1}^{n} X_{i\alpha}, \tag{1}$$

describes the semi-degree output of the vertex i, and the same sum in relation to the columns, i.e.:

$$C_i = \sum_{\alpha=1}^{n} X_{\alpha i}, \tag{2}$$

describes the semi-degree input of the vertex i.

In this manner, each non-negative integer matrix X has a corresponding directed pseudograph with n vertices, described by the set of two dimensional, integer and non negative vectors of semi-degrees $\overline{d_1} = (R_1, C_1), \ldots, \overline{d_n} = (R_n, C_n)$, for which:

$$\sum_{i=1}^{n} R_i = \sum_{i=1}^{n} C_i \tag{3}$$

The first two relations between the elements of pseudograph adjacency matrix and the semi-degrees of its vertices can be considered as a system of constraints with unknown elements and known vectors of the vertices semi-degrees. The statement (3) is the condition of consistency between the systems of constraints (1) and (2), which will be

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

connected in one system and can be written in the following way:

$$\begin{array}{ccccccccccc}
x_{11} & + & x_{12} & + & x_{13} & + & \cdots & + & x_{1n} & = & R_1 \\
+ & & + & & + & & & & + & & \\
x_{21} & + & x_{22} & + & x_{23} & + & \cdots & + & x_{2n} & = & R_3 \\
+ & & + & & + & & & & + & & \\
\vdots & & \vdots & & \vdots & & & & \vdots & & \vdots \\
+ & & + & & + & & & & + & & \\
x_{n1} & + & x_{n2} & + & x_{n3} & + & \cdots & + & x_{nn} & = & R_n \\
\| & & \| & & \| & & & & \| & & \\
C_1 & & C_2 & & C_3 & & \cdots & & C_n & &
\end{array} \tag{4}$$

The system of constraints (4) for given $d_j = (R_j, C_j)$ and $n > 1$ has many solutions in the general case. Truly, the difference between the number of unknown variables and the number of independent constraints is equal to $(n-1)^2$. The set $P\left(\overline{d_1}, \ldots, \overline{d_n}\right)$ of non negative integer solution is limited $0 \leq X_{ij} \leq \min(R_i, C_j)$, for $1 \leq (i,j) \leq n$ and can be described by $(n-1)^2$ independent integer parameters. Each solution can be represented by the labelled graph.

From the theory of system of linear constraints, we know that the solution of heterogeneous system of linear constraints can be presented as a sum of some specific solution of the heterogeneous system and the solution of the homogeneous system corresponding with them.

In order to find the connection networks which has been described by (4), we have to resolve the system of constraints N. We can transform the N to the homogeneous system J:

$$N = \left\{ \begin{array}{ccccccccccc}
x_{11} & + & x_{12} & + & x_{13} & + & \cdots & + & x_{1n} & = & R_1 \\
\vdots & & \vdots & & \vdots & & & & \vdots & & \vdots \\
x_{n1} & + & x_{n2} & + & x_{n3} & + & \cdots & + & x_{nn} & = & R_n \\
x_{11} & + & x_{21} & + & x_{31} & + & \cdots & + & x_{n1} & = & C_1 \\
\vdots & & \vdots & & \vdots & & & & \vdots & & \\
x_{1n} & + & x_{2n} & + & x_{3n} & + & \cdots & + & x_{nn} & = & C_n
\end{array} \right.$$

$$\Downarrow$$

$$J = \left\{ \begin{array}{ccccccccccc}
x_{11} & + & x_{12} & + & x_{13} & + & \cdots & + & x_{1n} & -(R_1) & = 0 \\
\vdots & & \vdots & & \vdots & & & & \vdots & \vdots & \vdots \\
x_{n1} & + & x_{n2} & + & x_{n3} & + & \cdots & + & x_{nn} & -(R_n) & = 0 \\
x_{11} & + & x_{21} & + & x_{31} & + & \cdots & + & x_{n1} & -(C_1) & = 0 \\
\vdots & & \vdots & & \vdots & & & & \vdots & \vdots & \vdots \\
x_{1n} & + & x_{2n} & + & x_{3n} & + & \cdots & + & x_{nn} & -(C_n) & = 0
\end{array} \right.$$

For the multibus systems described in **[5, 6]** which have n vertices and are represented by the bipartition graph we can reduce the number of equations in the corresponding

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

systems of constraints. To achieve that the following three conditions have to be fulfilled:

(1) there are no loops in the network i.e.: $x_{ij} = 0$ for $i = j$ and $i, j = 1, ..., n$,

(2) there are no connections between the nodes of the same type in the bipartition graph which represent the multibus system. If the vertices $v_1, v_2, \ldots, v_m$ represent the computational nodes, where $m < n$, and the vertices $v_{m+1}, v_{m+2}, \ldots, v_n$ represent the communication buses, then the above condition for the matrix X elements can be written in the following form:

$$\begin{matrix} x_{1\,1} & \ldots & x_{1\,m} \\ \vdots & & \vdots \\ x_{m\,1} & \ldots & x_{m\,m} \end{matrix} = \begin{matrix} x_{m+1\,m+1} & \ldots & x_{m+1\,n} \\ \vdots & & \vdots \\ x_{n\,m+1} & \ldots & x_{n\,n} \end{matrix} = 0,$$

(1) connections in the graph which represent the multibus system are undirected i.e:$x_{1\,m+1} = x_{m+1\,1},\ x_{1\,n} = x_{n\,1},\ \ldots, x_{m\,n} = x_{n\,m}$.

Taking into consideration points 1, 2, 3, the system of constraints can be reduced to the following equivalent forms:

$$\begin{cases} x_{1\,m+1} + \ldots + x_{1\,n} = R_1 \\ \vdots \quad\quad\quad\quad \vdots \\ x_{m\,m+1} + \ldots + x_{m\,n} = R_m \\ x_{1\,m+1} + \ldots + x_{m\,m+1} = C_1 \\ \quad\quad\quad\quad\quad\quad \vdots \\ x_{1\,n} + \ldots + x_{m\,n} = C_m \end{cases} \Leftrightarrow \begin{cases} x_{m+1\,1} + \ldots + x_{m+1\,m} = R_{m+1} \\ \vdots \quad\quad\quad\quad \vdots \\ x_{n\,1} + \ldots + x_{n\,m} = R_n \\ x_{m+1\,1} + \ldots + x_{n\,1} = C_{m+1} \\ \quad\quad\quad\quad\quad\quad \vdots \\ x_{m+1\,m} + \ldots + x_{n\,m} = C_n \end{cases}$$

where $R_1 = R_{m+1}, \ldots, R_m = R_n$and $C_1 = C_{m+1}, \ldots, C_m = C_n$.

2 The base of the solutions in the area $\{0, 1\}$

If we modify the TSS [**1, 7, 8**] algorithms, then we will use them to search solutions base in the area $\{0, 1\}$. In [**1, 7, 8**] the algorithms based on the use of the Contejean-Devie condition (SOL01 algorithm) were proposed. Below, we describe the algorithm which is based on the incrementation properties of the algorithm TSS.

Let us consider the class SLHDE (system of linear homogeneous Diophantine equations) in the following form:

$$S = \begin{cases} 1x_1 + 0x_2 \ldots + 0x_q & 1x_{q+1}0x_{q+2} \ldots 0x_{2q} - 1x_{2q+1} = 0 \\ 0x_1 + 1x_2 \ldots + 0x_q & 0x_{q+1}1x_{q+2} \ldots 0x_{2q} - 1x_{2q+1} = 0 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots & \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ 0x_1 + 0x_2 \ \ldots + 1x_q & 0x_{q+1}0x_{q+2} \ldots 1x_{2q} - 1x_{2q+1} = 0 \end{cases}$$

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

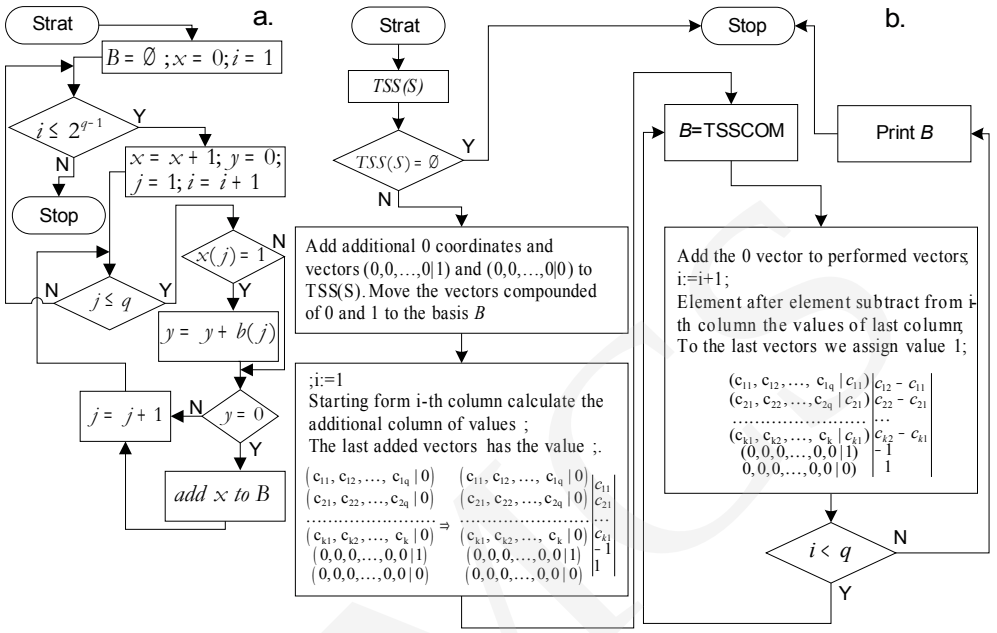


Fig. 1. Algorithm structure: a) TSS{01}, b) Algorithm 3.

The structure of such a system has the form: to the two identity matrices with the dimension $q \times q$, ordered directly one by one, the column filled with -1 is added. Such SLHDE are corresponding to system in following form:

$$S = \begin{cases} x_1 & \leq & 1 \\ \dots & \dots & \dots \\ x_q & \leq & 1 \end{cases}.$$

The use of the TSS algorithm to resolve such a system gives the solutions a set consistent with the SLHDE base and consisting of 2^q vectors. These vectors have t he following properties: a) the first of the q elements completes each other i.e. if i element of the first part is equal to 1 (0), then $q + i$-th component of the second part is equal to 0 (1), and the last coordinate of such a solution is always equal to 1; b) the values of the first of the q component are increasing from 0 to $2^q - 1$, i.e. they have all possible combinations 0 and 1. The TSS of such system is consistent with the base of all solutions and is compounded with 16 vectors.

The presented properties of the solution set for such a type of SLHDE allow to build the algorithm for search of the base of all solutions for the general form of SLHDE on the set $\{0, 1\}$, which is characterized by minimal memory complexity. It is based on the incrementation properties of TSS, and on the properties a) presented above. The full searching of the variants is realized, which is more effective than searching from method [**9**] and there is no need to use additional variables. For more detailed analysis,

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

the following form of the systems of equations and inequalities will be considered:

$$S' = \begin{cases} S & \leq & 1 \\ x_1 & \leq & 1 \\ \dots & \dots & \dots \\ x_q & \leq & 1 \end{cases}.$$

Introducing the additional variables and modifying the system to form the equations system we have the following SLHDE:

$$S" = \begin{cases} S & & & & & & & & & \\ 1 & 0 & \dots & 0 & 1 & \dots & 0 & -1 & = & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & \dots & 1 & 0 & \dots & 1 & -1 & = & 0 \end{cases}.$$

From the incrementation properties of the TSS creation it follows that TSS of a given system is identical to TSS of the system in following form:

$$S_1 = \begin{cases} 1 & 0 & \dots & 0 & 1 & \dots & 0 & -1 & = & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & \dots & 1 & 0 & \dots & 1 & -1 & = & 0 \\ S & & & & & & & & & \end{cases}.$$

From that follows that there is no need to determine TSS of the subsystem prefacing the system S (top part), because the set is known. On the other hand, because the equations of the system S are complemented with zeros, then only the first of the q coordinates of TSS of the top subsystem has influence on the final form of the TSS of the whole system. As the way to determine these coordinates is known from the properties a), then during determination of TSS there is no need to use additional variables or to save the TSS vectors of the top subsystem. TSS for the whole system can be determined by browsing the sequence of all TSS vectors of top subsystem, and saves only solutions base. During this, different types of browsing can be used.

According to the above speculations and using the natural order of the binary vectors we will describe data structures of the algorithm and the algorithm itself.

Let x be a q-dimensional binary vector which saves the actual values of the SLHDE solution vector, B is the base of all the solutions set of S in the area $\{0,\ 1\}$, $b(j)$ are the columns of the matrix A of the system S. For the above data structure, the algorithm has the form presented in Fig 1.a:

$$TSS\{01\}\ (b_1, b_2, \dots, b_q, p)$$

Input: $b_1, b_2, \dots b_q$ - the columns of matrix A of the system $A \cdot x = 0$, a p - their dimension.

Output:B - base of the all solution set of the system $A \cdot x = 0$ in the area $\{0, 1\}$.

The time complexity of a given algorithm determines the following theorem:

Theorem 1. *The time complexity of the algorithm TSS{01} is proportional to* $pq \cdot 2^{q-1}$.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

Note that both algorithms (TSS{01} and SOL01) are dedicated to any number area, because they do not depend on the values of the system factors. Besides, these algorithms can be applied without modification in the case of heterogeneous systems **[2, 9]**.

Taking into consideration the evaluation of time complexity of the $SOL01$ or $TSS\{01\}$algorithms, the second one is more effective. The memory complexity of both algorithms is identical and equals: $\max(pq, |B|\, q)$.

3 Algorithm 3

Incrementation properties of the TSS algorithm allowed to prepare the algorithm to resolve the SLHDE on the set $\{0, 1\}$ and proved its fullness i.e. possibility to create base of all solutions set **[8, 9, 6]**. However, the experiments show that generation time depends on the form of resolved systems. The experiments show that the base creation time can be dozens of times longer when the upper and bottom parts of the coefficient matrix have been exchanged.

The detailed analysis of the above fact showed, that it can be used to make a new base creation algorithm which will be use only two additional variables instead of p.

Let us consider SLHDE in the form:

$$S = \begin{cases} a_{11}x_1 + \ldots + a_{1q}x_q & = & 0 \\ a_{21}x_1 + \ldots + a_{2q}x_q & = & 0 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots & \ldots & \ldots \\ a_{p1}x_1 + \ldots + a_{pq}x_q & = & 0 \end{cases}$$

Where: $a_{ij} \in Z$ (the ring of integer numbers), $i = 1, 2, \ldots p, j = 1, 2, \ldots q$. We have to find the base of solutions for a given SLHDE on the set $\{0, 1\}$.

The algorithm structure is presented in Fig. 1.b and the TSSCOM procedure can be described as follows.

TSSCOM

The vector combinations are made in the same way as it was done before in the case of the TSS algorithm using the last value column, which is separated from the others with vertical lines in the example in Fig. 1.b. If the obtained TSS has vectors, whose coordinates are compounded with 0 and 1, then we move the vector (vectors) to the base B according to the joining rules and we remove them from TSS. The minimization of the current TSS is made in two stages (stage a. and stage b.)

(1) The vectors from the current base (if it is not empty) are compared with the current TSS. If the vector from the base is bigger than that from the current TSS then it is changed to the vector from TSS which is removed from TSS. If the vector from TSS is bigger than that from the base, then we remove it from TSS and next go to point 2.

(2) The vectors from TSS are compared. If one of the vectors is bigger than the other one, then their complement in relation to the last coordinate of

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

both vectors is calculated. If the obtained vectors are in the same relation as source vectors, then the bigger vector is removed from the current TSS. Note that we do not need to build complement openly. Indeed, if for a step $j,\ j \in [2,q]$ the values $_d$ and d', placed in the last column (which is placed between the vertical lines) and the relation in which the complements of vectors $x = (a_1, a_2, \ldots, a_q)$ and $y = (b_1, b_2, \ldots, b_q)$ are calculated then for $a_i \geq b_i,\ i = 1, 2, \ldots, j$ the complements will be in the same relations, if $d - a_i \geq d' - b_i$ or $d - d' \geq a_i - b_i$ for $i = 1, 2, \ldots, j$. As a result, if $x \geq y$ then during the comparison the differences $c = d - d'$ and $c_i = a_i - b_i$ can be used. If $c \geq c_i$ for all $i \in [1, j]$ the vector x is removed from the current TSS.

As a result, we have the following vectors:

$$
\begin{array}{l}
(c_{11}, c_{12}, \ldots, c_{1q}|c_{11}) \\
(c_{21}, c_{22}, \ldots, c_{2q}|c_{21}) \\
\ldots\ldots\ldots\ldots\ldots\ldots \\
(c_{k1}, c_{k2}, \ldots, c_{k}|c_{k1}) \\
(0, 0, 0, \ldots, 0, 0|1)
\end{array}
$$

If all the vectors with coordinates bigger than 1 were removed from the set we obtain the set of solutions which contains all base vectors of the solutions in the set $\{0, 1\}$. The justification for the use of the given algorithm follows from its incrementation properties.

4 Algorithm MTSS{0,1}

The idea of the proposed algorithm is to reduce the set of possible solutions of the system S. The previous algorithms generate a set consisting of all possible combinations of vectors with the coordinates $0 - 1$, length q, where q is the number of variables in the system of equations S (e.g. $q = 20$ the set consists of 2^{20} vectors) Note that for a given system S, the factors matrix has the size: $(A + B) \times (A \cdot B + 1)$:

$$
\left[\begin{array}{ccccc}
11..1 & 00..0 & ... & 00..0 & p_1 \\
00..0 & 11..1 & ... & 00..0 & p_2 \\
 & & ... & & \\
00..0 & 00..0 & ... & 11..1 & p_A \\
10..0 & 10..0 & ... & 10..0 & m_1 \\
01..0 & 01..0 & ... & 01..0 & m_2 \\
 & & ... & & \\
00..1 & 00..1 & ... & 00..1 & m_B
\end{array}\right].
$$

where: A is the number of vertices (CPU) in the system, B - the number of buses. The algorithm searches the solutions to the "upper" part, that is, equations from 1 to A, and checks that among these solutions are those of the "bottom" part. The solutions are formed in the following way: for the first equation solution the vectors

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

$x_1 + x_2 + \ldots + x_B = p_1$ are formed, which have a length B, and they have a set of components from the set $\{0, 1\}$; solutions vectors for the second equation $x_{B+1} + x_{B+2} + \ldots + x_{2B} = p_2$ $(\ldots)$ until the last of the equation of the "upper" part $x_{(A-1)B+1} + x_{(A-1)B+2} + ... + x_{AB} = p_A$. All the solution vectors from the "upper" part havea length B and are the vectors $0-1$. The next step is to create all combinations of solutions which will give vectors of a length A · B. To these vectors we must add the last component – the 1 – it corresponds to the factors $[p_1, p_2, p_3, \ldots, p_A]$. In this manner, all the possible solutions of the "upper" part have been achieved. It should be checked which of these solutions are the solution for equation from the "lower" part of the system.

In the case of this algorithm, the number of generated vectors and the number of checks are limited from $2^{A \cdot B}$, to $\prod_1^A C_B^{p_i}$.

The problem of obtaining solutions for each equation of the "upper" part can be solved by a single generation of all combinations of vectors of the component $\{0, 1\}$ and the length B. Then, the vectors are stored in a matrix, and their number in these vectors is compared to the absolute value of the factor p_i for each equation. The algorithms to generate the B-bit words which contain a certain number of ones can be also used.

Diagram of the algorithm is shown in Fig. 2.

The variable C determines the number of possible solutions (the size of M), which then have to be checked for solvability of the lower part of S. The set M is generated on the basis of the Cartesian product of each of individual sets of solutions for equations of the upper part of S. The check whether the element of the set (vector) M is the solution of lower part, can be made by multiplying a vector by a factor matrix of the lower part. If you receive a zero matrix, then the vector is the solution.

5 Coclusions

In this paper the new method for multibus network created with given parameters has been proposed. These networks can be used to construct the computational systems (to connect CPU), as well as for the synthesis of the topology for wireless networks. The algorithms described in Chapters 2 and 3 are classical algorithms that have been adapted to search the topology with given parameters. In Chapter 4, a new algorithm for searching topology for the multibus network was proposed. Experimental studies show that this algorithm is much less sensitive to the order of the equations in the system of constraints which describes the bus system. MTSS algorithm also generates a smaller set of potential solutions (already for 4 nodal topology with 6 connections MTSS achieved 10 times smaller set of solutions than the classical algorithms), which reduces the final selection topology . A lot of research has been made to find the new possibilities of the use of the presented method. For example, based on [**10**], the matrix model for the wireless network convergence can be converted to the algebraic model resolved by the presented method. Future works will concentrate on developing

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

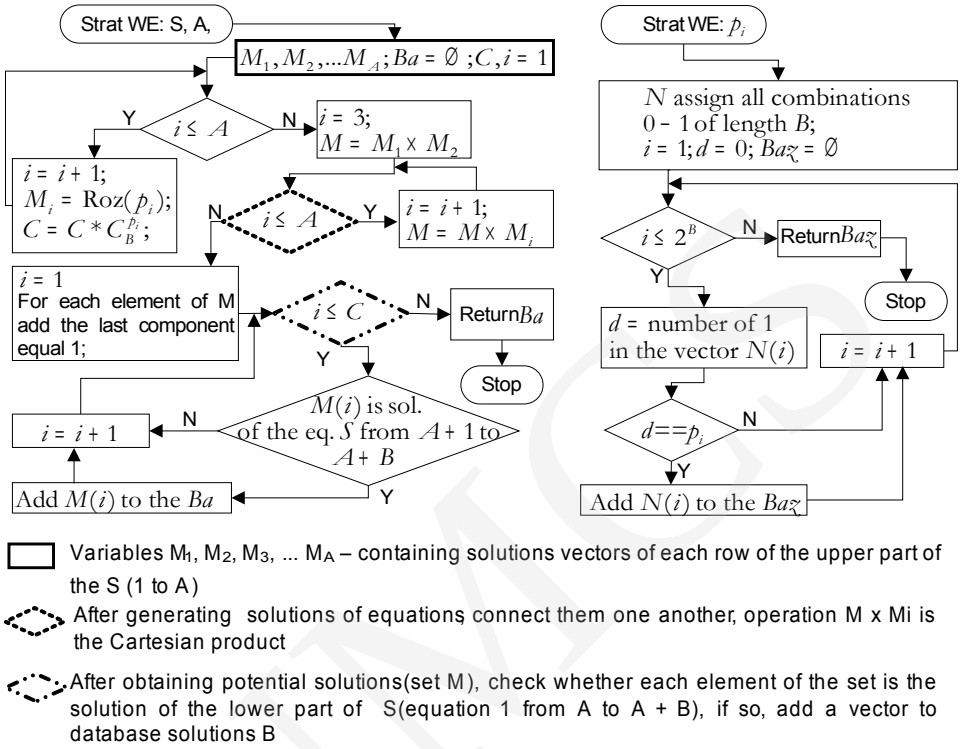


Fig. 2. Diagram of the algorithm MTSS{0,1}

methods to select the final topology from the obtained set of solutions taking indeterminacy into account. A lot of acceptable solutions and more than one criterion of evaluation cause indeterminacy. This approach will make extensive use of it in the case of the designing systems characterized by a complexity that describe topologies and a variability of function conditions.

References

[1] Red. Stroinski M., Studium rozwoju sieci szerokopasmowej dla wojewodztwa podkarpackiego na lata 2007-2013, Osrodek Wydawnictw Naukowych Instytut Chemii Bioorganicznej Polska Akademia Nauk, Oddzial w Poznaniu; ISBN 978-83-7314-128-5.

[2] Rajzer G., Kombinatornaja matematika, Mir, Rosja (1965).

[3] Bronsztejn I.N., Siemendiajew K.A., Musiol G., Muhlig H., Nowoczesne kompendium matematyki, Wydawnictwo Naukowe PWN, Warszawa (2004).

[4] Atallah M. J. (Editor), Algorithms and Theory of Computation Handbook, CRC Press, Boca Raton (2002).

[5] Hajder M., Bolanowski M., Connectivity analysis in the computational systems with distributed Communications in the multichannel environment, Polish Journal of Environmental Studies 17 (2A) (2008): 14.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 26/11/2026 16:20:12

[6] Byczkowska-Lipinska L., Filipiak-Karasinska A., Dymora P., Method of Coverage Improvement in Wireless Regional Networks, Proccedings of the Third European Conference on the Use of Modern Information and Communication Technologies, ECUMICT, Gent, 13-14 March (2008): 57.

[7] Krivoi S., A Criteria of compatibility systems of linear diophantine constraints, Lecture Notes in Computer Science. Spring. Verlag N 2328 (2002): 264.

[8] Krivoi S., Hajder M., Bagryi R., Application of algorithms for solving linear constrain systems, Algebra, Logic and Cibernetics. Proceedings. Irkutsk (2004): 165.

[9] Krivoi S., Grzywacz W., Hajder M., Algorithm for constructing the solutions basis in the set {0,1}, Conf. "Algebra, Logic, Cybernetics", Irkutsk (2004): 167.

[10] Hajder M., Byczkowska-Lipinska L., Bolanowski M., Analysis and synthesis of a computational system with reconfigurable multichannel connections, X International PhD Workshop OWD'2008, 18–21 October (2008).

UMCS