2

The Network Operator Method for Search of the Most Suitable Mathematical Equation

Askhat Diveev[1] and Elena Sofronova[2]
[1]*Institution of Russian Academy of Sciences Dorodnicyn Computing Centre of RAS,*
[2]*Peoples' Friendship University of Russia*
Russia

1. Introduction

For many applied and research problems it is necessary to find solution in the form of mathematical equation. These problems are the selection of function at approximation of experimental data, identification of control object model, control synthesis in the form of state space coordinates function, the inverse problem of kinetics and mathematical physics, etc. The main method to receive mathematical equations for solution of these problems consists in analytical transformations of initial statement formulas of the problem. A few problems have the exact analytical solution, therefore mathematicians use various assumptions, decomposition, and special characteristics of solutions. Usually mathematicians set the form of mathematical equation, and the optimal parameters are found using numerical methods and PC. Such methods as the least-square method have been applied to the problems of approximation for many years (Kahaner D. et al., 1989).

Recently the neural networks have been used to solve complex problems when the mathematical equation cannot be found analytically. The structure of any neural network is also given within the values of parameters or weight coefficients. In problems of function approximation and the neural network training the form of mathematical equation is set by the researcher, and the computer searches for optimum values of parameters in these equations (Callan, 1999; Demuth et al., 2008).

In 1992 a new method of genetic programming was developed. It allows to solve the problem of search of the most suitable mathematical equation. In genetic programming mathematical equations are represented in the form of symbol strings. Each symbol string corresponds to a computation graph in the form of a tree. The nodes of this graph contain operations, and the leaves contain variables or parameters (Koza, 1992, 1994; Koza, Bennett et al., 1999 & Koza, Keane et al., 2003).

Genetic programming solves the problems by applying genetic algorithm. To perform the crossover it is necessary to find symbol substrings that correspond to brunches of trees. The analysis of symbol strings increases the operating time of the algorithm. If the same parameter or variable is included in the required mathematical equation several times, then to solve the problem effectively the genetic programming needs to crossover the trees so that the leaves contain no less than the required number of parameters or variables.

Limitations of the genetic programming revealed at the solution of the problem of suitable mathematical equation search, have led to creation of the network operator.

In this work we introduce a new data structure which we called a network operator. Network operator is a directed graph that contains operations, arguments and all information for calculations of mathematical equation.

Network operator method was used for the problems of control synthesis (Diveyev & Sofronova, 2008; Diveev, 2009; Diveev & Sofronova, 2009a,b).

2. Program notations of mathematical equations

Mathematical equations consist of variables, parameters, unary and binary operations that form four constructive sets.

Set of variables

$$\mathrm{X} = (x_1,\ldots,x_N),\ x_i \in \mathrm{R}^1,\ i=\overline{1,N}. \tag{1}$$

Set of parameters

$$\mathrm{Q} = (q_1,\ldots,q_P),\ q_i \in \mathrm{R}^1,\ i=\overline{1,P}. \tag{2}$$

Unary operations set

$$\mathrm{O}_1 = (\rho_1(z)=z, \rho_2(z),\ldots,\rho_W(z)). \tag{3}$$

Binary operations set

$$\mathrm{O}_2 = (\chi_0(z',z''),\ldots,\chi_{V-1}(z',z'')). \tag{4}$$

Unary operations set must have an identity operation

$$\rho_1(z) = z. \tag{5}$$

Binary operations must be commutative

$$\chi_i(z',z'') = \chi_i(z'',z'),\ i=\overline{0,V-1}, \tag{6}$$

associative

$$\chi_i((z',z''),z''') = \chi_i(z',(z'',z''')),\ i=\overline{0,V-1}, \tag{7}$$

and have a unit element

$$\exists e_i \ \Rightarrow\ \chi_i(e_i,z) = z,\ i=\overline{0,V-1}. \tag{8}$$

A program notation of mathematical equation is a notation of equation with the help of constructive sets (1) – (4).

3. Graphic notations of mathematical equations

To present mathematical equation as a graph we use a program notation. Let us enlarge the program notation by additional unary identity operation $\rho_1(z) = z$ and binary operation with a unit element $\chi_i(e_i, z) = z$. These operations do not influence the result of calculation but they set a definite order of operations in the notation, so that binary operations have unary operations or unit elements as their arguments, and unary operations have only binary operations, parameters or variables as their arguments.

A graphic notation of mathematical equation is a notation of binary operation that fulfills the following conditions:

a. binary operation can have unary operations or unit element of this binary operation as its arguments;
b. unary operation can have binary operation, parameter or variable as its argument;
c. binary operation cannot have unary operations with the same constants or variables as its arguments.

Any program notation can be transformed into a graphic notation.

4. Network operator of mathematical expression

To construct a graph of the mathematical expression we use a graphic notation. The graphic notation can be transformed into the graph if unary operations of mathematical expression correspond to the edges of the graph, binary operations, parameters or variables correspond to the nodes of the graph.

Suppose that in graphic notation we have a substring where two unary operations are arguments to binary operation $\ldots\chi_k(\rho_l(\ldots),\rho_m(\ldots))\ldots$ This substring is presented as a graph on Fig. 1

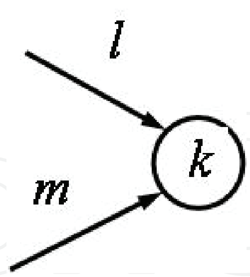


Fig. 1. The graph for substring $\ldots\chi_k(\rho_l(\ldots),\rho_m(\ldots))\ldots$

Suppose we have a substring where binary operation is an argument to unary operation $\ldots\rho_k(\chi_l(\ldots))\ldots$. This substring is presented as a graph on Fig. 2.

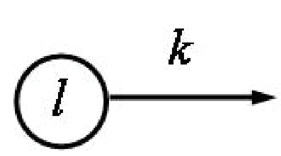


Fig. 2. The graph for substring $\ldots\rho_k(\chi_l(\ldots))\ldots$

Let us have a substring where parameter or variable is an argument to unary operation $\ldots\rho_k(a)\ldots$, where a is an argument or parameter of mathematical equation, $a \in X \cup Q$. The graph for this substring is presented on Fig. 3.

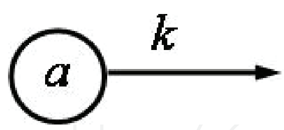


Fig. 3. The graph for substring $\ldots\rho_k(a)\ldots$, $a \in X \cup Q$

If graphic notation contains a substring where binary operation with a unit element is an argument to unary operation $\ldots\rho_k(\chi_l(\rho_m(\ldots),0))\ldots$ We do not depict this unit elements and the node has only one incoming edge as shown on Fig. 4.

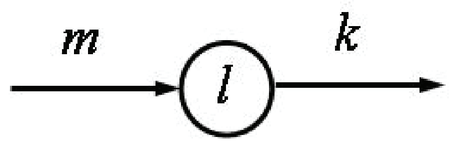


Fig. 4. The graph for substring $\ldots\rho_k(\chi_l(\rho_m(\ldots),0))\ldots$

5. Properties of network operators

Network operator is a directed graph that has the following properties:

a. graph has no loops;
b. any nonsource node has at least one edge from the source node;
c. any non sink node has at least one edge to sink node;
d. every source node corresponds to the element from the set of variables X or the set of parameters Q;
e. every node corresponds to binary operation from the set of binary operations O_2;
f. every edge corresponds to unary operation from the set of unary operations O_1.

To calculate mathematical expression we have to follow certain rules:

a. unary operation is performed only for the edge that comes out from the node with no incoming edges;
b. the edge is deleted from the graph once the unary operation is performed;
c. the binary operation in the node is performed right after the unary operation of the incoming edge is performed;
d. the calculation is terminated when all edges are deleted from the graph.

To construct most of mathematical expressions we use the sets of unary and binary operations that are given in Table 1 and Table 2.

Consider the construction of the network operator for the following mathematical equation

$$y = x_1 + \sin(x_1) + q_1 x_1 e^{-x_2} \ .$$

$\rho_1(z)=z$	$\rho_{13}(z)=\arctan(z)$
$\rho_2(z)=\begin{cases}\varepsilon^{-1}, \text{ if } \lvert z\rvert>\dfrac{1}{\sqrt{\varepsilon}}\\ z^2, \text{ otherwise}\end{cases}$	$\rho_{14}(z)=\begin{cases}\dfrac{\operatorname{sgn}(z)}{\varepsilon}, \text{ if } \lvert z\rvert>\dfrac{1}{\sqrt[3]{\varepsilon}}\\ z^3, \text{ otherwise}\end{cases}$
$\rho_3(z)=-z$	$\rho_{15}(z)=\begin{cases}\sqrt[3]{\varepsilon}, \text{ if } \lvert z\rvert<\varepsilon\\ \sqrt[3]{z}, \text{ otherwise}\end{cases}$
$\rho_4(z)=\operatorname{sgn}(z)\sqrt{\lvert z\rvert}$	$\rho_{16}(z)=\begin{cases}z, \text{ if } \lvert z\rvert<1\\ \operatorname{sgn}(z), \text{ otherwise}\end{cases}$
$\rho_5(z)=\begin{cases}\dfrac{\operatorname{sgn}(z)}{\varepsilon}, \text{ if } \lvert z\rvert<\varepsilon\\ \dfrac{1}{z}, \text{ otherwise}\end{cases}$	$\rho_{17}(z)=\operatorname{sgn}(z)\ln(\lvert z\rvert+1)$
$\rho_6(z)=\begin{cases}\varepsilon^{-1}, \text{ if } z>-\ln(\varepsilon)\\ e^z, \text{ otherwise}\end{cases}$	$\rho_{18}(z)=\begin{cases}\dfrac{\operatorname{sgn}(z)}{\varepsilon}, \text{ if } \lvert z\rvert>-\ln(\varepsilon)\\ \operatorname{sgn}(z)\left(e^{\lvert z\rvert}-1\right), \text{ otherwise}\end{cases}$
$\rho_7(z)=\begin{cases}\ln(\varepsilon), \text{ if -}\ln\lvert z\rvert>\dfrac{1}{\varepsilon}\\ \ln\lvert z\rvert, \text{ otherwise}\end{cases}$	$\rho_{19}(z)=\begin{cases}0, \text{ if } \lvert z\rvert>-\ln(\varepsilon)\\ \operatorname{sgn}(z)e^{-\lvert z\rvert}, \text{ otherwise}\end{cases}$
$\rho_8(z)=\begin{cases}\operatorname{sgn}(z), \text{ if } \lvert z\rvert>-\ln(\varepsilon)\\ \dfrac{1-e^{-z}}{1+e^{-z}}, \text{ otherwise}\end{cases}$	$\rho_{20}(z)=\begin{cases}1, \text{ if } z>\varepsilon\\ 0, \text{ if } z<0\\ \dfrac{3z^2}{\varepsilon^2}-\dfrac{2z^3}{\varepsilon^3}, \text{ otherwise}\end{cases}$
$\rho_9(z)=\begin{cases}1, \text{ if } z\geq 0\\ 0, \text{ otherwise}\end{cases}$	$\rho_{21}(z)=\begin{cases}1, \text{ if } z>\dfrac{\varepsilon}{2}\\ -1, \text{ if } z<-\dfrac{\varepsilon}{2}\\ \dfrac{3z}{\varepsilon^2}-\dfrac{4z^3}{\varepsilon^3}, \text{ otherwise}\end{cases}$
$\rho_{10}(z)=\operatorname{sgn}(z)$	$\rho_{22}(z)=\begin{cases}0, \text{ if } \lvert z\rvert>-\ln(\varepsilon)\\ e^{-\lvert z\rvert}, \text{ otherwise}\end{cases}$
$\rho_{11}(z)=\cos(\mathrm{z})$	$\rho_{23}(z)=\begin{cases}-\dfrac{\operatorname{sgn}(z)}{\varepsilon}, \text{ if } \lvert z\rvert>\dfrac{1}{\sqrt[3]{\varepsilon}}\\ z-z^3, \text{ otherwise}\end{cases}$
$\rho_{12}(z)=\sin(z)$	$\rho_{24}(z)=\begin{cases}\dfrac{\varepsilon}{1+\varepsilon}, \text{ if } z>-\ln(\varepsilon)\\ \dfrac{1}{1+e^{-z}}, \text{ otherwise}\end{cases}$

Table 1. Unary operations

Operation	Unit element
$\chi_0(z',z'') = z' + z''$	0
$\chi_1(z',z'') = z'z''$	1
$\chi_2(z',z'') = \max\{z',z''\}$	$-\frac{1}{\varepsilon}$
$\chi_3(z',z'') = \min\{z',z''\}$	$\frac{1}{\varepsilon}$
$\chi_4(z',z'') = z' + z'' - z'z''$	0
$\chi_5(z',z'') = \mathrm{sgn}(z'+z'')\sqrt{(z')^2 + (z'')^2}$	0
$\chi_6(z',z'') = \mathrm{sgn}(z'+z'')(\lvert z'\rvert + \lvert z''\rvert)$	0
$\chi_7(z',z'') = \mathrm{sgn}(z'+z'')\max\{\lvert z'\rvert,\lvert z''\rvert\}$	0

Table 2. Binary operations

First we set parentheses to emphasize the arguments of functions. Then using Table 1 and Table 2 we find appropriate operations and replace functions by operations

$$y = x_1 + \sin(x_1) + q_1 x_1 e^{-x_2} = \left((x_1 + \sin(x_1)) + q_1 x_1 e^{-x_2}\right) = \chi_0\left((x_1 + \sin(x_1)), q_1 x_1 e^{-x_2}\right)$$
$$= \chi_0\left(\chi_0(x_1, \sin(x_1)), q_1 x_1 e^{-x_2}\right) = \chi_0\left(\chi_0(x_1, \rho_{12}(x_1)), \chi_1(q_1 x_1, \rho_6(-x_2))\right)$$
$$= \chi_0\left(\chi_0(x_1, \rho_{12}(x_1)), \chi_1(\chi_1(q_1, x_1), \rho_6(\rho_3(x_2)))\right).$$

As a result we obtain a program notation of mathematical equation

$$y = \chi_0\left(\chi_0(x_1, \rho_{12}(x_1)), \chi_1(\chi_1(q_1, x_1), \rho_6(\rho_3(x_2)))\right).$$

We can see that this program notation does not meet the requirements to graphic notation. These requirements are necessary for further construction of the graph. According to the definition of the network operator binary operations correspond to the nodes of the graph, unary operations correspond to the edges, thus binary and unary operations must be arguments of each other in the graphic notation of mathematical equation.

When a binary operation has as its argument in program notation then we cannot construct the graph, because there is no edge, in other words no unary operation, between two nodes.

To meet the requirements for graphic notation let us introduce additional unary identity operations. For example in the given program notation we have a substring

$$y = \chi_0(\chi_0(\ldots), \chi_1(\ldots)).$$

Here binary operation has two binary operations as its arguments. It does not satisfy condition «a» of graphic notation. If we use additional identity operation, then we have

$$y = \chi_0\left(\rho_1\left(\chi_0(\ldots)\right), \rho_1\left(\chi_1(\ldots)\right)\right).$$

Unary identity operation $\rho_1(z)$ does not change the value of argument and this operation is necessary for the construction of the graph by graphic notation.

Since graphic notation should contain binary operations with unary operations as their arguments then we additional unary identity operations. We get

$$y = \chi_0\left(\rho_1\left(\chi_0\left(\rho_1(x_1), \rho_{12}(x_1)\right)\right), \ \rho_1\left(\chi_1\left(\rho_1\left(\chi_1\left(\rho_1(q_1), \rho_1(x_1)\right)\right), \rho_6\left(\rho_3(x_2)\right)\right)\right)\right).$$

This notation is not a graphic one, because it does not satisfy condition «b» of graphic notation. Here unary operation has unary operation as its argument

$$\ldots\rho_6\left(\rho_3(x_2)\right)\ldots$$

We use additional binary operation with a unit element as its second argument, for example $\chi_0(\ldots,0)$. According to Table 2 binary operation $\chi_0(z',z'')$ is addition. A unit element for addition is 0 and it does not influence the result of calculation. Thus we get

$$\ldots\rho_6\left(\chi_0\left(\rho_3(x_2),0\right)\right)\ldots$$

We obtain the graphic notation of mathematical equation

$$y = \chi_0\left(\rho_1\left(\chi_0\left(\rho_1(x_1), \rho_{12}(x_1)\right)\right), \ \rho_1\left(\chi_1\left(\rho_1\left(\chi_1\left(\rho_1(q_1), \rho_1(x_1)\right)\right), \rho_6\left(\chi_0\left(\rho_3(x_2),0\right)\right)\right)\right)\right).$$

This notation does not satisfy condition «c» of graphic notation, because it contains a substring where binary operation has two unary operations with the same variable as its arguments

$$\ldots\chi_0\left(\rho_1(x_1), \rho_{12}(x_1)\right)\ldots$$

We add a binary operation with a unit element and a unary identity operation to the substring

$$\ldots\chi_0\left(\rho_1\left(\chi_0\left(\rho_1(x_1),0\right)\right), \rho_{12}(x_1)\right)\ldots$$

As a result we get the following notation of mathematical equation

$$y = \chi_0\left(\rho_1\left(\chi_0\left(\rho_1\left(\chi_0\left(\rho_1(x_1),0\right)\right), \rho_{12}(x_1)\right)\right), \ \rho_1\left(\chi_1\left(\rho_1\left(\chi_1\left(\rho_1(q_1), \rho_1(x_1)\right)\right), \rho_6\left(\chi_0\left(\rho_3(x_2),0\right)\right)\right)\right)\right).$$

This notation has all properties of graphic notation and we can construct the graph of equation by this notation. To construct the graph we use the rules presented on Fig. 1 - 4. The graph is shown on Fig. 5.

Fig. 5 shows the numeration of nodes on the top of each node in the graph. We see that the numbers of the nodes where the edges come out from are less than the numbers of nodes where the edges come in. Such numeration is always possible for directed graphs without loops.

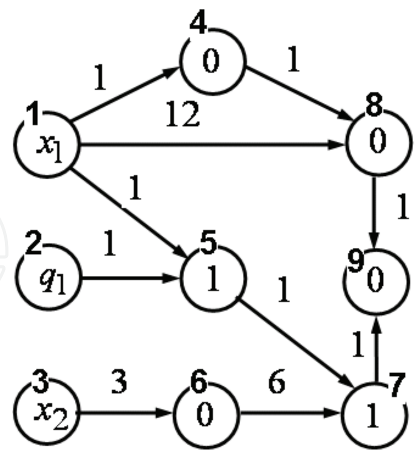


Fig. 5. Graph of mathematical equation

To calculate the mathematical equation which is presented as a graph we use additional vector of nodes $\mathbf{z}$ for storage of intermediate results. Each element of vector $\mathbf{z}$ is associated with the definite node in the graph. Initially elements of vector z_i that are associated with the source nodes have the values of variables and parameters. For example for the graph presented at Fig. 5 we have

$$\mathbf{z} = \begin{bmatrix} z_1 & \dots & z_9 \end{bmatrix}^T,$$

where 9 is the number of nodes in the graph. For the source nodes we set $z_1 = x_1$, $z_2 = q_1$, $z_3 = x_2$. Values of other elements z_i are equal to the unit elements for binary operations. As a result we get an initial value of vector of nodes

$$\mathbf{z} = \begin{bmatrix} x_1 & q_1 & x_2 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}^T.$$

In the given example we use addition and multiplication. Unit element for addition is 0 and for multiplication is 1.

According to the rules of calculation, we calculate unary operation that corresponds to the edge that comes out from the node that has no incoming edges. For the edge (i,j) node i has no incoming edges at the moment. Unary operation ρ_k corresponds to the edge (i,j). Binary operation χ_l corresponds to the node j. Then we perform the following calculations

$$z_j = \chi_l\left(z_j, \rho_k\left(z_i\right)\right), \tag{9}$$

where z_j in the right part of the equation is the value on the previous step.

After calculation of (9) we delete the edge (i,j) from the graph.

If we numerate the nodes so that the number of the node where the edge comes out from is less than the number of the node that it comes in, then the calculation can be done just following the numbers of the nodes.

For the given example we have the following steps:

- edge (1,4), $z_4 = \chi_0(\rho_1(z_1), z_4) = x_1 + 0 = x_1$;
- edge (1,8), $z_8 = \chi_0(\rho_{12}(z_1), z_8) = \sin(x_1) + 0 = \sin(x_1)$;
- edge (1,5), $z_5 = \chi_1(\rho_1(z_1), z_5) = x_1 1 = x_1$;
- edge (2,5), $z_5 = \chi_1(\rho_1(q_1), z_5) = q_1 x_1$;
- edge (3,6), $z_6 = \chi_0(\rho_3(x_2), z_6) = -x_2 + 0 = -x_2$;
- edge (4,8), $z_8 = \chi_0(\rho_1(z_4), z_8) = x_1 + \sin(x_1)$;
- edge (5,7), $z_7 = \chi_1(\rho_1(z_5), z_7) = q_1 x_1 1 = q_1 x_1$;
- edge (6,7), $z_7 = \chi_1(\rho_6(z_6), z_7) = e^{-x_2} q_1 x_1$;
- edge (7,9), $z_9 = \chi_0(\rho_1(z_7), z_9) = e^{-x_2} q_1 x_1 + 0 = e^{-x_2} q_1 x_1$;
- edge (8,9), $z_9 = \chi_0(\rho_1(z_8), z_9) = x_1 + \sin(x_1) + e^{-x_2} q_1 x_1$.

When the calculations on the edge (8,9) are performed we obtain the result of initial mathematical expression.

Nodes 8 and 9 in the graph can be united since binary operations are associative and commutative. A reduced graph of mathematical equation is given on Fig. 6.

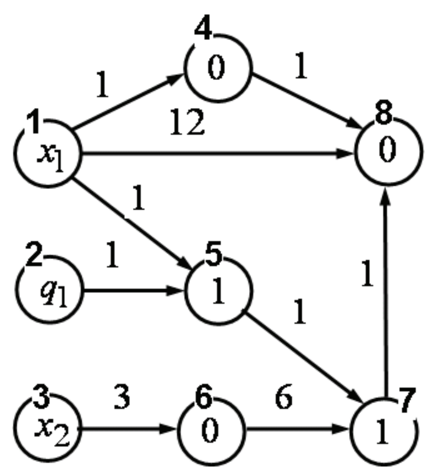


Fig. 6. Reduced graph of mathematical equation

The results of calculation for graphs presented on Fig. 5 and Fig. 6 are the same.

The result of calculation will not change if we unite two nodes that are linked by the edge that corresponds to unary identical operation and the edges that are linked to that nodes do not come in or out from the same node.

To construct the graph of mathematical equation we need as many nodes as the sum of parameters, variables and binary operations in its graphic notation. This number is enough for construction but not minimal.

The result of calculation will not change if to the sink node of the graph we add an edge with a unary identical operation and a node with binary operation and a unit element. An enlarged graph for given example is shown on Fig. 7.

A directed graph constructed form the graphic notation of mathematical equation is a network operator. One network operator can be associated with several mathematical equations. It depends on the numbers of sink nodes that are set by the researcher. In the given example if we numerate the sink nodes with numbers 7, 8, 9 then we will get three mathematical equations

$$y_1 = z_7 = e^{-x_2} q_1 x_1 ,$$

$$y_2 = z_8 = x_1 + \sin(x_1),$$

$$y_3 = z_9 = x_1 + \sin(x_1) + e^{-x_2} q_1 x_1 .$$

This feature of the graphic notation allows using the network operator for presentation of vector functions.

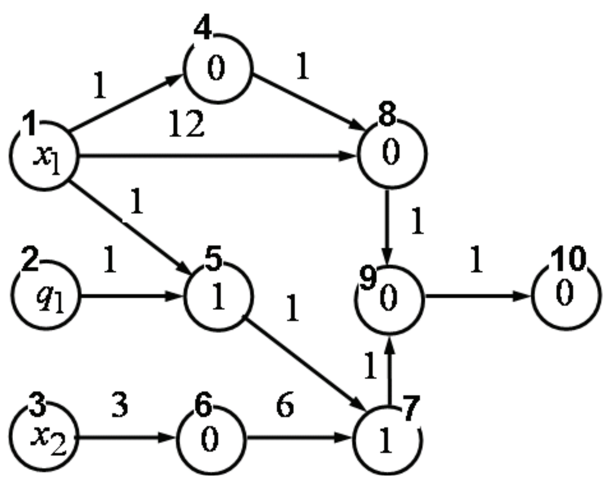


Fig. 7. Enlarged graph of mathematical equation

6. Network operator matrices

To present a network operator in the PC memory we use a network operator matrix (NOM). NOM is based on the incident matrix of the graph $\mathbf{A} = [a_{ij}]$, $a_{ij} \in \{0,1\}$, $i,j = \overline{1,L}$, where L is the number of nodes in the graph.

If we replace diagonal elements of the incident matrix with numbers of binary operations that correspond to appropriate nodes and nonzero nondiagonal elements with numbers of unary operations, we shall get NOM $\mathbf{\Psi} = [\psi_{ij}]$, $i,j = \overline{1,L}$.

For the network operator shown on the Fig. 6 we have the following NOM

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

NOM for the graph on Fig. 6 is the following

$$\mathbf{\Psi}=\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 12 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

NOM $\mathbf{\Psi}=\left[\psi_{i,j}\right]$, $i,j=\overline{1,L}$ is upper-triangular because of the numeration of nodes. NOM is not enough for calculation of mathematical equation since it does not contain information about parameters and variables. This information is kept in the initial values of vector of nodes

$$\mathbf{z}=\left[z_1 \quad \dots \quad z_L\right]^T . \tag{10}$$

Then the calculation of mathematical equation can be done by

$$z_j=\chi_{\psi_{jj}}\left(\rho_{\psi_{ij}}\left(z_i\right),z_j\right),\ \text{if}\ \psi_{ij}\neq 0\ ,\ i=\overline{1,L-1}\ ,\ j=\overline{i+1,L}\ . \tag{11}$$

To calculate the mathematical equation by its NOM we need to look through all rows consequently, $i=\overline{1,L-1}$. In each row i we consider the elements that follow the diagonal element, $j=\overline{i+1,L}$. If among them we find nonzero element then we perform calculation according equation (11).

For the given NOM we get

$$\mathbf{z}=\left[x_1 \quad q_1 \quad x_2 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0\right]^T ,$$

$\psi_{1,4}=1$, $\psi_{4,4}=0$, $z_4=\chi_0\left(\rho_1\left(z_1\right),z_4\right)=x_1+0=x_1$;
$\psi_{1,5}=1$, $\psi_{5,5}=1$, $z_5=\chi_1\left(\rho_1\left(z_1\right),z_5\right)=x_1 1=x_1$;
$\psi_{1,8}=12$, $\psi_{8,8}=0$, $z_8=\chi_0\left(\rho_{12}\left(z_1\right),z_8\right)=\sin\left(x_1\right)+0=\sin\left(x_1\right)$;
$\psi_{2,5}=1$, $\psi_{5,5}=1$, $z_5=\chi_1\left(\rho_1\left(q_1\right),z_5\right)=q_1x_1$;
$\psi_{3,6}=3$, $\psi_{6,6}=0$, $z_6=\chi_0\left(\rho_3\left(x_2\right),z_6\right)=-x_2+0=-x_2$;
$\psi_{4,8}=1$, $\psi_{8,8}=0$, $z_8=\chi_0\left(\rho_1\left(z_4\right),z_8\right)=x_1+\sin\left(x_1\right)$;
$\psi_{5,7}=1$, $\psi_{7,7}=1$, $z_7=\chi_1\left(\rho_1\left(z_5\right),z_7\right)=q_1x_1 1=q_1x_1$;
$\psi_{6,7}=6$, $z_7=\chi_1\left(\rho_6\left(z_6\right),z_7\right)=e^{-x_2}q_1x_1$;
$\psi_{7,8}=1$, $\psi_{8,8}=0$, $z_8=\chi_0\left(\rho_1\left(z_7\right),z_8\right)=e^{-x_2}q_1x_1+x_1+\sin\left(x_1\right)$.

7. Variations of network operators

Similar network operators are network operators that satisfy the following conditions:

a. have the same source nodes;
b. have the same constructive sets.

Alike network operators are similar network operators that have equal numbers of nodes.

Network operators of alike structure are alike network operators that differ in unary and binary operations.

Variation of network operator is the change of network operator that leads to a similar network operator.

Simple variation of network operator is a variation that cannot be presented as a complex of other variations.

Simple variations of network operator are given in Table 3.

Number of simple variation	Simple variation
0	replacement of unary operation on the edge
1	replacement of binary operation in the node
2	addition of the edge with a unary operation
3	deletion of the edge if the node where this edge comes in has at least one more incoming edge
4	Increase of the node number
5	Decrease of the node number
6	addition of the node with a binary operation and incoming edge with unary operation
7	deletion of the sink node with incoming edge if this edge is single.

Table 3. Simple variations of network operator

Structural variation of network operator is a variation that changes the set of edges of network operator.

Structural variations change the incident matrix of the graph. In the Table 3 structural variations are 2 – 5. Variations 0, 1 do not change the incident matrix and lead to network operators of alike structure.

A complete network operator is a network operator in which we cannot perform variation 2.

A complete network operator contains L nodes in which the number of source nodes is $N+P$, maximum number of edges is equal to

$$|\mathrm{C}| = \frac{(L-N-P)(L+N+P-1)}{2} . \tag{12}$$

If we apply variation 2, addition of an edge, to any network operator, then we can construct a full network operator which is alike initial network operator.

If we apply variation 3, deletion of an edge, to the complete network operator, then we can construct any alike network operator.

Any variation of network operator can be performed by a finite number of simple variations.

An eigen variation of network operator is a variation that does not change the number of nodes in the network operator.

In the Table 3 eigen variations are 0 – 5.

Any eigen variation of the network operator can be performed by a finite number of simple eigen variations.

To present any simple variation we use a variation vector

$$\mathbf{w} = \left[w_1 \, w_2 \, w_3 \, w_4 \right]^T , \tag{13}$$

where w_1 is the number of variation from Table 3, w_2, w_3, w_4 are elements that integer values depend on the number of variation.

Values of elements of variation vector are given in Table 4. In case the values of elements are not defined they can take any values. For example when $w_1 = 1$ element w_2 can keep the number of the node where this edge comes in $w_2 = w_3$.

Variation of network operator is presented as

$$\tilde{\mathbf{\Psi}} = \mathbf{w} \circ \mathbf{\Psi} ,$$

where $\mathbf{\Psi}$ is the NOM before variation is performed, $\tilde{\mathbf{\Psi}}$ is the NOM after variation was performed.

Number of variation w_1	Number of the node where the edge comes out	Number of the node where the edge comes in	Number of unary operation	Number of binary operation
0	w_2	w_3	w_4	-
1	-	w_3	-	w_4
2	w_2	w_3	w_4	-
3	w_2	w_3	-	-
4	w_2	-	-	-
5	w_2	-	-	-
6	w_2	-	w_3	w_4
7	-	w_3	-	-

Table 4. Elements of variation vector

Consider examples of variations of network operator. We have a network operator that describes mathematical equation

$$y = \frac{q_1 x_1^3}{\sqrt[3]{x_1^2 + x_2^2}} .$$

Network operator matrix for the given equation is

$$\mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Suppose we have a variation vector $\mathbf{w} = [2\ 4\ 6\ 2]^T$. Element $w_1 = 2$ shows that we perform addition of the edge. According to Table 3 a new edge should come out from the node 4, come in the node 6 and have unary operation 2.

As a result we have NOM

$$\tilde{\mathbf{\Psi}} = \mathbf{w} \circ \mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

NOM $\tilde{\mathbf{\Psi}}$ corresponds to the following mathematical equation

$$y = \frac{q_1 x_1^3}{\sqrt[3]{x_1^2 + x_2^2 + \left(q_1 x_1^3\right)^2}}.$$

Suppose variation vectors $\mathbf{w}^1 = [0\ 4\ 7\ 11]^T$ and $\mathbf{w}^2 = [0\ 6\ 7\ 1]^T$ are given. The first component of these vectors $w_1 = 0$ shows the replacement of unary operation on the edge. The second and the third components show the edge between the nodes. The first vector points to the edge $(4,7)$, the second – to the edge $(6,7)$. The forth element contains the number of new unary operation. According to Table 1 this operation for vector $\mathbf{w}^1$ is $\rho_{11}(z) = \cos(z)$, for vector $\mathbf{w}^2$ is $\rho_1(z) = z$. As a result we obtain NOM

$$\tilde{\mathbf{\Psi}} = \mathbf{w}^2 \circ \mathbf{w}^1 \circ \mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 11 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

NOM $\tilde{\mathbf{\Psi}}$ corresponds to

$$y = \cos\left(q_1 x_1^3\right)\sqrt[3]{x_1^2 + x_2^2} .$$

Consider the examples of improper variations that change the number of nodes in the network operator. We have a variation vector $\mathbf{w} = \left[6\ 4\ 7\ 0\right]^T$. Number of variation $w_1 = 6$ shows that we add the node with binary operation $w_4 = 0$ and an outcoming edge with unary operation $w_3 = 7$. After variation we obtain the NOM

$$\tilde{\mathbf{\Psi}} = \mathbf{w} \circ \mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

This NOM corresponds to the graph with two sink nodes and it presents at least two mathematical equations

$$y_1 = \frac{q_1 x_1^3}{\sqrt[3]{x_1^2 + x_2^2}},\ y_2 = \ln\left|q_1 x_1^3\right| .$$

Let us given variation vectors $\mathbf{w}^1 = \left[3\ 6\ 7\ 0\right]^T$ and $\mathbf{w}^2 = \left[5\ 7\ 0\ 0\right]^T$. In the first vector $w_1^1 = 3$, that is why we delete the edge between nodes $w_2^1 = 6$ and $w_3^1 = 7$. In the second vector $w_1^2 = 5$, and we delete the node $w_2^2 = 7$ with its incoming edge. As a result we have

$$\tilde{\mathbf{\Psi}} = \mathbf{w}^2 \circ \mathbf{w}^1 \circ \mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 14 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

This NOM corresponds to two mathematical equations

$$y_1 = q_1 x_1^3,\ y_2 = \sqrt[3]{x_1^2 + x_2^2} .$$

Since we have changed the graph we obtain two mathematical equations. Network operator is presented on Fig. 8.

Performance of variations is not always possible. If variation cannot be done then it is omitted. For example we have a variation vector $\mathbf{w} = \left[0\ 4\ 6\ 2\right]^T$. The first component shows variation 0, replacement of unary operation. However there is no edge between nodes $w_2 = 4$ and $w_3 = 6$, that is why this variation is not performed and NOM is not changed

$$\mathbf{\Psi} = \mathbf{w} \circ \mathbf{\Psi} .$$

For proper variations variation $w_1 = 3$ is not performed if one of the following conditions is fulfilled:

- edge (w_2, w_3) is absent;
- there are no other incoming edges to the node where the edge (w_2, w_3) comes in;
- there are no other outcoming edges from the node where the edge (w_2, w_3) comes out.

A structural distance between two similar network operators is a minimal number of single variations that should be performed to obtain one network operator from the other. A structural distance between network operator $\mathbf{\Psi}^1$ and network operator $\mathbf{\Psi}^2$ is equal to a structural distance between network operators $\mathbf{\Psi}^2$ and $\mathbf{\Psi}^1$.

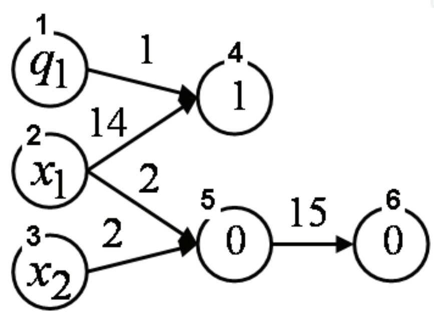


Fig. 8. Network operator after variations

8. Search of optimal mathematic equation

Let us formulate the problem of search of optimal mathematical equation $\mathbf{g}(\mathbf{x},\mathbf{q})$. It is necessary to find mathematical equation $\mathbf{g}(\mathbf{x},\mathbf{q})$, that provides minimums of object functions with restrictions

$$\min f_i(\mathbf{x},\mathbf{g}(\mathbf{x},\mathbf{q})),\ i=\overline{0,D}, \tag{14}$$

$$f_j(\mathbf{x},\mathbf{g}(\mathbf{x},\mathbf{q})) \le 0,\ j=\overline{D+1,K}, \tag{15}$$

where $\mathbf{x}$ is a vector of variables, $\mathbf{q}$ is a vector of parameters.

The solution to the problem (14), (15) is a mathematical equation $\tilde{\mathbf{g}}(\mathbf{x},\tilde{\mathbf{q}})$. Mathematical equation $\mathbf{g}(\mathbf{x},\mathbf{q})$ is a function

$$\mathbf{g}(\mathbf{x},\mathbf{q}) = \left[g_1(\mathbf{x},\mathbf{q}) \ldots g_M(\mathbf{x},\mathbf{q})\right]^T. \tag{16}$$

If $D \ge 1$, then the solution of the problem is a Pareto set

$$\Pi = \left\{\tilde{\mathbf{g}}^{i_k}(\mathbf{x},\tilde{\mathbf{q}}):\ k=\overline{1,s}\right\},$$

where $\forall \mathbf{g}^k(\mathbf{x},\mathbf{q}) \notin \Pi\ \ \exists \tilde{\mathbf{g}}^j(\mathbf{x},\tilde{\mathbf{q}}) \in \Pi$, $\mathbf{f}^0\left(\tilde{\mathbf{g}}^j(\mathbf{x},\tilde{\mathbf{q}})\right) \le \mathbf{f}^0\left(\mathbf{g}^k(\mathbf{x},\mathbf{q})\right)$, $0 \le j \le D$, $\mathbf{f}^0\left(\mathbf{g}(\mathbf{x},\mathbf{q})\right) = \left[f_0\left(\mathbf{g}(\mathbf{x},\mathbf{q})\right) \ldots f_D\left(\mathbf{g}(\mathbf{x},\mathbf{q})\right)\right]^T$, $\mathbf{f}^0\left(\mathbf{g}^j(\mathbf{x},\mathbf{q})\right) \le \mathbf{f}^0\left(\mathbf{g}^k(\mathbf{x},\mathbf{q})\right)$, if $f_i\left(\mathbf{g}^j(\mathbf{x},\mathbf{q})\right) \le f_i\left(\mathbf{g}^k(\mathbf{x},\mathbf{q})\right)$, $i=\overline{0,D}$, and $\exists f_l\left(\mathbf{g}^j(\mathbf{x},\mathbf{q})\right) < f_l\left(\mathbf{g}^k(\mathbf{x},\mathbf{q})\right)$.

To solve the problem (14), (15) we need to define a finite set of network operators, and in this set find the best solution accoding to (14).

In the set of network operators we choose a basic network operator. Basic network operator corresponds to the basic NOM $\mathbf{\Psi}^0$ and the basic mathematical equation $\mathbf{g}^0\left(\mathbf{x},\mathbf{q}^0\right)$, where $\mathbf{q}^0 = \left[q_1^0 \dots q_P^0\right]^T$ is a vector of parameters.

Let us introduce a finite ordered set of variation vectors

$$\mathrm{W} = \left(\mathbf{w}^1,\dots,\mathbf{w}^l\right), \tag{17}$$

where l is a cardinal number of W, $\mathbf{w}^i = \left[w_1^i \dots w_4^i\right]^T$ is a variation vector, $i=\overline{1,l}$.

To construct the set of network operators we use a basic matrix $\mathbf{\Psi}^0$ and all possible sets W of variation vectors.

9. Genetic algorithm for method of variations of basic solution

Consider genetic algorithm that searches both structure and parameters of mathematical equation.

Initially we set the basic solution

$$\mathbf{\Psi}^0 = \left[\psi_{ij}^0\right],\ i,j=\overline{1,L}\,. \tag{18}$$

We generate the ordered sets of variation vectors

$$\mathrm{W}^i = \left(\mathbf{w}^{i,1},\dots,\mathbf{w}^{i,l}\right),\ i=\overline{1,H}, \tag{19}$$

$$\mathbf{w}^{i,j} = \left[w_1^{i,j} \quad w_2^{i,j} \quad w_3^{i,j} \quad w_4^{i,j}\right]^T,\ i=\overline{1,H},\ j=\overline{1,l}, \tag{20}$$

where H is a number of possible solutions in the population.

We generate bit strings for parameters

$$\mathbf{s}^i = \left[s_1^i \dots s_{P(c+d)}^i\right]^T,\ i=\overline{1,H}, \tag{21}$$

where P is the number of parameters, c is the number of bits for the integer part of the value, and d is the number of bits for the fractional part.

For each chromosome $\left(\mathrm{W}^i,\mathbf{s}^i\right)$, $1 \le i \le H$, we define the values of object functions. We construct NOM using $W^i = \left(\mathbf{w}^{i,1},\dots,\mathbf{w}^{i,l}\right)$ and $\mathbf{\Psi}^0$

$$\mathbf{\Psi}^i = \mathbf{w}^{i,l} \circ \dots \circ \mathbf{w}^{i,1} \circ \mathbf{\Psi}^0\,. \tag{22}$$

We present parametrical part of the chromosome $\mathbf{s}^i = \left[s_1^i \dots s_{P(c+d)}^i\right]^T$, $1 \le i \le H$, as a vector of parameters. We present a bit string $\mathbf{s}^i$ from the Gray code to the binary code

$$\mathbf{b}^i = \left[b_1^i \dots b_{P(c+d)}^i\right]^T, \tag{23}$$

$$b_j^i = \begin{cases} s_j^i, \text{ if } (j-1) \bmod (c+d) = 0 \\ s_j^i \oplus b_{j-1}^i, \text{ otherwise} \end{cases}, \; j = \overline{1, P(c+d)}. \tag{24}$$

From the binary code we obtain the vector of parameters

$$\mathbf{q}^i = \left[q_1^i \ldots q_P^i \right]^T, \tag{25}$$

$$q_k^i = \sum_{j=1}^{c+d} 2^{c-j} b_{j+(k-1)(c+d)}^i, \; k = \overline{1, P}. \tag{26}$$

To estimate each possible solution $\mathbf{\Psi}^i$ and $\mathbf{q}^i$ we use parameter which is called a distance to a Pareto set.

A distance to a Pareto set is a number of possible solutions that are better in terms of Pareto than the current solution.

For each solution $\left(\mathrm{W}^j, \mathbf{s}^j\right)$, $1 \le j \le H$ we find mathematical equation $\mathbf{g}^j\left(\mathbf{x}, \mathbf{q}^j\right)$ and calculate the values of object functions (14)

$$\mathbf{f}^j = \left[f_0^j \ldots f_D^j \right]^T, \tag{27}$$

In the set we find the number of possible solutions that are better than the solution $\left(\mathrm{W}^j, \mathbf{s}^j\right)$

$$\Lambda_j = \sum_{i=1}^{H} \lambda_i\left(\mathbf{f}^j\right), \tag{28}$$

where

$$\lambda_i\left(\mathbf{f}^j\right) = \begin{cases} 1, \text{ if } \mathbf{f}^i \le \mathbf{f}^j \\ 0, \text{ otherwise} \end{cases}. \tag{29}$$

To construct new solutions we perform genetic operations of selection, crossover and mutation.

We randomly choose two solutions $\left(\mathrm{W}^{i_1}, \mathbf{s}^{i_1}\right)$, $\left(\mathrm{W}^{i_2}, \mathbf{s}^{i_2}\right)$ and perform a crossover with probability

$$p_c = \max\left\{ \frac{1+\gamma\Lambda_{i_1}}{1+\Lambda_{i_1}}, \frac{1+\gamma\Lambda_{i_2}}{1+\Lambda_{i_2}} \right\}, \tag{30}$$

where γ is a given crossover parameter, $0 < \gamma < 1$.

After crossover is performed in two points k_p, k_s we obtain four new solutions $\left(\mathrm{W}^{H+1}, \mathbf{s}^{H+1}\right)$, $\left(\mathrm{W}^{H+2}, \mathbf{s}^{H+2}\right)$, $\left(\mathrm{W}^{H+2}, \mathbf{s}^{H+3}\right)$, $\left(\mathrm{W}^{H+4}, \mathbf{s}^{H+4}\right)$

$$\mathrm{W}^{H+1} = \mathrm{W}^{i_1}, \tag{31}$$

$$\mathbf{s}^{H+1} = \left[s_1^{i_1} \dots s_{k_p}^{i_1} s_{k_p+1}^{i_2} \dots s_{P(c+d)}^{i_2} \right]^T , \tag{32}$$

$$\mathrm{W}^{H+2} = \mathrm{W}^{i_2} , \tag{33}$$

$$\mathbf{s}^{H+2} = \left[s_1^{i_2} \dots s_{k_p}^{i_2} s_{k_p+1}^{i_1} \dots s_{P(c+d)}^{i_1} \right]^T , \tag{34}$$

$$\mathrm{W}^{H+3} = \left(\mathbf{w}^{i_1,1}, \dots, \mathbf{w}^{i_1,k_s-1}, \mathbf{w}^{i_2,k_s}, \dots, \mathbf{w}^{i_2,l} \right), \tag{35}$$

$$\mathbf{s}^{H+3} = \left[s_1^{i_1} \dots s_{k_p}^{i_1} s_{k_p+1}^{i_2} \dots s_{P(c+d)}^{i_2} \right]^T , \tag{36}$$

$$\mathrm{W}^{H+4} = \left(\mathbf{w}^{i_2,1}, \dots, \mathbf{w}^{i_2,k_s-1}, \mathbf{w}^{i_1,k_s}, \dots, \mathbf{w}^{i_1,l} \right), \tag{37}$$

$$\mathbf{s}^{H+4} = \left[s_1^{i_2} \dots s_{k_p}^{i_2} s_{k_p+1}^{i_1} \dots s_{P(c+d)}^{i_1} \right]^T . \tag{38}$$

For each new solution $\left(\mathrm{W}^{H+i},\mathbf{s}^{H+i}\right)$, $i=\overline{1,4}$, we perform a mutation with probability $p_m \in [0,1]$. We find the points of mutation m_s, m_p for both parts of new solutions. In the new chromosome $\left(\mathrm{W}^{H+i},\mathbf{s}^{H+i}\right)$, $1 \le i \le 4$, we randomly generate a variation vector $\mathbf{w}_{m_s}^{H+i}$ with structural and parametric parts. For each new solutions we calculate the functions (14)

$$\mathbf{f}^{H+i} = \left[f_0^{H+i} \dots f_D^{H+i} \right]^T , \; i=\overline{1,4} . \tag{39}$$

For a new solution j we find the distance to Pareto set Λ_{H+j} according to (28). Then we find the solution with a maximum distance to Pareto set

$$\Lambda_{i_+} = \max\left\{ \Lambda_i, i=\overline{1,H} \right\}, \tag{40}$$

where i_+ is a number of solution with maximum distance to Pareto set.

We compare new solution to the solution that has maximum distance to Pareto set

$$\Lambda_{H+j} < \Lambda_{i_+} . \tag{41}$$

If (41) is fulfilled then we replace the solution with a maximum distance by the first new solution $\mathrm{W}^{i_+} = \mathrm{W}^{H+j}$, $\mathbf{s}^{i_+} = \mathbf{s}^{H+j}$, $\mathbf{f}^{i_+} = \mathbf{f}^{H+j}$, and recalculate the distances for all solutions in the set.

These steps are performed for each new possible solution $\left(\mathrm{W}^{H+j},\mathbf{s}^{H+j}\right)$, $j=\overline{1,4}$.

The steps are repeated starting from the selection of possible solutions. After several given E iterations, where E is called epoch, we change basic solution $\mathbf{\Psi}^0$.

As a new basic solution we can take the solution that has minimum of function

$$\tilde{\mathbf{f}}^{i_-} = \min\left\{ \sqrt{\sum_{j=0}^{D} \left(\tilde{f}_j^i \right)^2} \right\}, \tag{42}$$

where $i\_$ is the number of new basic solution, $\tilde{\mathbf{f}}^i = \left[\tilde{f}_0^i \ldots \tilde{f}_D^i\right]^T$, $i = \overline{1,H}$, $\tilde{f}_j^i = \frac{f_j^i - f_j^-}{f_j^+ - f_j^-}$, $j = \overline{0,D}$, $f_j^+ = \max\left\{f_j^i, i = \overline{1,H}\right\}$, $f_j^- = \min\left\{f_j^i, i = \overline{1,H}\right\}$, $j = \overline{0,D}$.

For the more rapid search we use a subset of elite solutions $\left(\mathrm{W}^{i_e}, \mathbf{s}^{i_e}\right)$, $1 \le i_e \le H$. In this subset we calculate the values of functional after each variation $\mathbf{f}^{i_e}\left(\mathbf{\Psi}^{i_e,k}, \mathbf{s}^{i_e}\right)$,

$$\mathbf{\Psi}^{i_e,k} = \mathbf{w}^{i_e,k} \circ \ldots \circ \mathbf{w}^{i_e,1} \circ \mathbf{\Psi}^0, \; 1 \le k \le d. \tag{43}$$

We find variation of the solution that leads to minimum distance to Pareto set $\Lambda_{i_e}\left(\mathbf{f}^{i_e,k}\right)$.

$$\Lambda_{i_e}\left(\mathbf{f}^{i_e,k\_}\right) = \min\left\{\Lambda_{i_e}\left(\mathbf{f}^{i_e,k}\right), k = \overline{1,d}\right\}, \tag{44}$$

where $k\_$ is the number of desired variation. Other variations for possible solution i_e are replaced by zeros. The calculation is terminated after given number of loops.

Consider an example. It is necessary to find inverse function for mathematical equation

$$x = \cos\left(y^2\right) + \sqrt[3]{y}.$$

The solution is presented in the form $y = g(x, \mathbf{q})$, where $\mathbf{q}$ is a vector of parameters.

After substitution of found mathematical equation $g(x, \mathbf{q})$ in initial function we should obtain the identity

$$x \equiv \cos\left(g^2(x, \mathbf{q})\right) + \sqrt[3]{g(x, \mathbf{q})}.$$

Let us set a finite number of points

$$T = \left\{x^j : j = \overline{1,S}\right\}$$

and define two object functions

$$f_1 = \sqrt{\sum_{j=1}^{S}\left(x^j - \cos\left(g^2\left(x^j, \mathbf{q}\right)\right) - \sqrt[3]{g\left(x^j, \mathbf{q}\right)}\right)^2} \to \min,$$

$$f_2 = \max_j\left\{\left|x^j - \cos\left(g^2\left(x^j, \mathbf{q}\right)\right) - \sqrt[3]{g\left(x^j, \mathbf{q}\right)}\right| : j = \overline{1,S}\right\} \to \min.$$

Note that for exact solution we have $f_1 = 0$ and $f_2 = 0$.

Let us choose the following basic solution

$$y = q_1 x + q_2,$$

where $q_1 = 1$, $q_2 = 1$.

Network operator for basic solution is presented on Fig. 9.

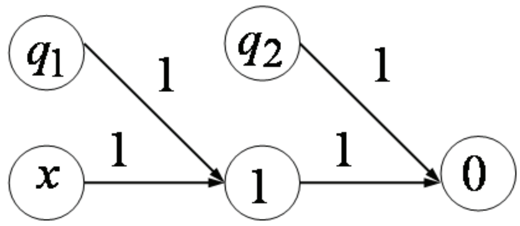


Fig. 9. Example of basic network operator

To construct a basic network operator we need 5 nodes, but if we want to enlarge the search space we add 3 nodes with addition operations and its unit elements. We get the network operator presented on Fig. 10.

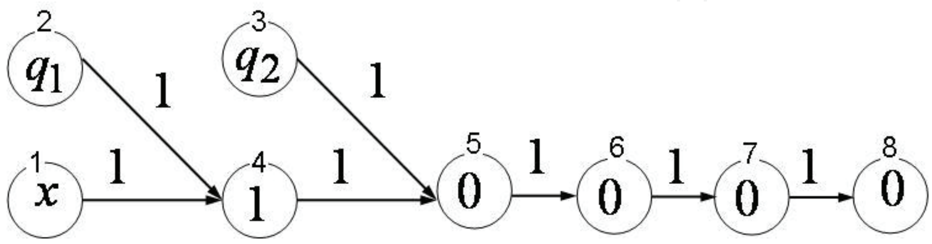


Fig. 10. An expanded basic network operator

NOM for graph shown on Fig. 10 is

$$\mathbf{\Psi}^0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

A genetic algorithm had the following values of parameters: number of chromosomes in initial population $H = 1024$, number of crossing couples in one generation 256, number of generations 128, number vectors of variations in one chromosome 8, number of generations between the change of basic solutions 22, number of elite chromosomes 8, probability of mutation $p_m = 0.8$, parameter for crossing $\gamma = 0.4$, number of parameters 2, number of bits for integer part $c = 2$, number of bits for fractional part $d = 6$, number of points $S = 11$. We obtained a Pareto set which is represented on Fig 11 and in the Table 5.

For example we take the solution no 310.

$$f_1\left(g^{310}(x,\mathbf{q})\right) = 0.32447275,\ f_2\left(g^{310}(x,\mathbf{q})\right) = 0.121647.$$

For this solution we have obtained the following values of parameters $q_1 = 3.14063$, $q_2 = 0.84375$.

The solution 310 is the network operator

$$\boldsymbol{\Psi}^{310}=\begin{bmatrix}0&0&0&1&0&0&0&0\\0&0&0&11&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&1&12&0&0&0\\0&0&0&0&0&12&15&0\\0&0&0&0&0&0&1&15\\0&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&1\end{bmatrix}.$$

or the function

$$g^{310}(x,\mathbf{q})=\sin\left(\sin\left(x\cos\left(q_1\right)\right)+q_2\right)\sqrt[3]{\left(\sin\left(x\cos\left(q_1\right)\right)+q_2\right)^2}.$$

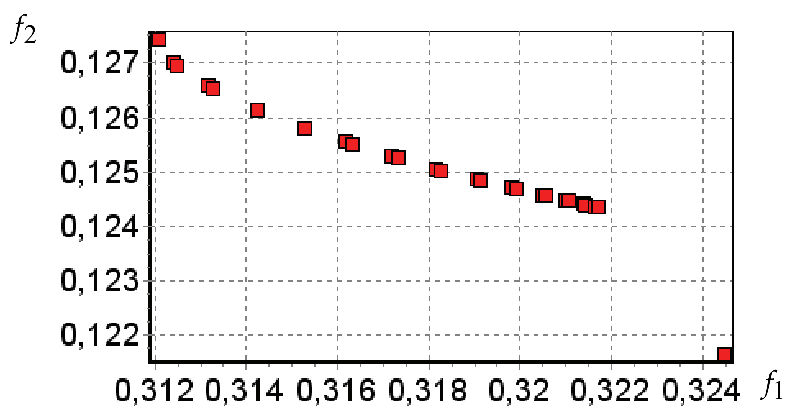


Fig. 11. A Pareto set

If we substitute our solution in initial mathematical expression then we have to obtain identity,

$$\tilde{x}=\cos\left(\left(g^{310}(x,\mathbf{q})\right)^2\right)+\sqrt[3]{\left(g^{310}(x,\mathbf{q})\right)}.$$

The graphs of the functions $\tilde{x}$ and identity function are represented on the Fig 12

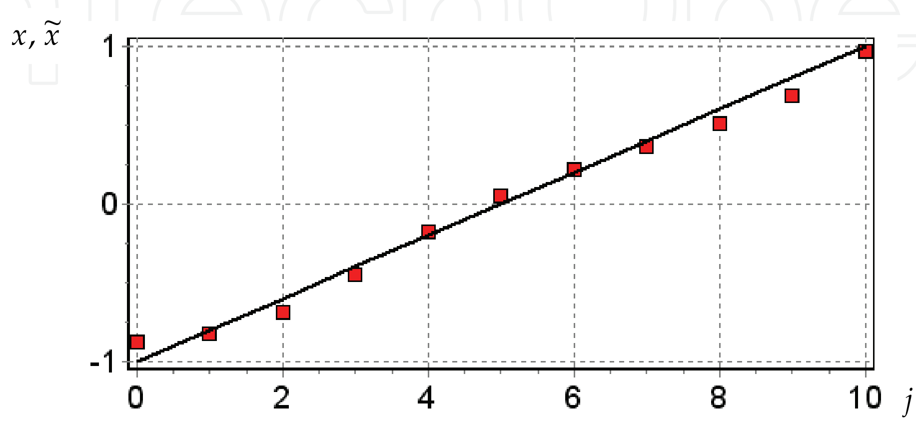


Fig 12. The graph of the function for the solution no 310

No	No of solution	f_1	f_2
1	27	0.31208313	0.12743371
2	469	0.3124092	0.12701286
3	36	0.31248842	0.12695655
4	366	0.31317121	0.12657987
5	492	0.3132807	0.12652959
6	122	0.31423625	0.12614932
7	412	0.31526313	0.12581255
8	173	0.31617553	0.12555099
9	868	0.31630362	0.12551643
10	472	0.31719342	0.12528834
11	494	0.31731617	0.12525837
12	18	0.31815508	0.12506176
13	106	0.31826894	0.1250361
14	624	0.31903468	0.12486921
15	54	0.31981169	0.12470895
16	180	0.31989995	0.12469127
17	132	0.32046985	0.12457951
18	560	0.32054244	0.12456557
19	539	0.32099647	0.12447971
20	205	0.32105204	0.12446936
21	288	0.321382	0.12440861
22	141	0.32141958	0.12440176
23	696	0.32161979	0.12436553
24	658	0.32163874	0.12436213
25	621	0.32170585	0.12435009
26	310	0.32447275	0.121647

Table 5.

10. Conclusion

In this work the new approach to the problem of automatic search of mathematical equations was considered. The researcher defines the sets of operations, variables and parameters. The computer program generates a number of mathematical equations that satisfy given restrictions. Then the optimization algorithm finds the structure of appropriate mathematical expression and its parameters. The approach is based on the new data structure the network operator.

If we replace the set of unary and binary operations in the network operator by the set of logic operations, then we can perform the search of the most suitable logic function (Alnovani et al. 2011).

11. Limitations & development

Presentation of the network operator as a matrix is limited by its dimension.

In the problems where mathematical equations have many variables and parameters, it is necessary to use big network operator matrices with many zero elements.

To exclude this limitation it is possible to divide one network operator with a considerable number of nodes into some small network operators. We receive the multilayer network operator and some matrices of smaller dimensions. Each layer of the network operator describes a part of mathematical equation.

Further development of the network operator is a creation of a special data structure for presentation of the network operator in memory of the computer. Such structure can be multilayered and provide effective parallel calculation.

12. References

Alnovani G.H.A., Diveev A.I., Pupkov K.A., Sofronova E.A. (2011) Control Synthesis for Traffic Simulation in the Urban Road Network, *Preprints of the 18-th IFAC World Congress.* Milan (Italy) August 28 – September 2. pp. 2196-2201.

Callan R. (1999) *The essence of neural networks*. The Prentice Hall Europe, 1999, ISBN 0-13-908732-X, 978-013-9087-32-5. 232 p.

Demuth, H.; Beale, M.; Hagan, M. (2008) *Neural Network Toolbox™ User's Guide*. The MathWorks, Inc. ISBN 0-9717321-0-8. 907 p.

Diveev A.I. (2009) A multiobjective synthesis of optimal control system by the network operator method. *Proceedings of international conference «Optimization and applications»* (OPTIMA) Petrovac, Montenegro, September 21-25, pp. 21-22.

Diveev A.I., Sofronova E.A. (2009) Numerical method of network operator for multi-objective synthesis of optimal control system, *Proceedings of Seventh International Conference on Control and Automation* (ICCA'09) Christchurch, New Zealand, December 9-11. pp. 701-708. ISBN 978-1-4244-4707-7.

Diveev A.I., Sofronova E.A. (2009) The Synthesis of Optimal Control System by the Network Operator Method, *Proceedings of IFAC Workshop on Control Applications of Optimization CAO'09*, 6 - 8 May, University of Jyväskylä, Agora, Finland.

Diveyev A.I., Sofronova E.A. (2008) Application of network operator method for synthesis of optimal structure and parameters of automatic control system, *Proceedings of 17-th IFAC World Congress*, Seoul, 05. - 12. July . pp. 6106 – 6113.

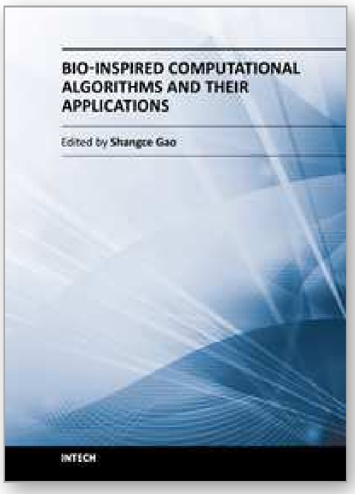
Kahaner, D.; Moler, C.; Nash, S. (1989) *Numerical methods and software* Prentice Hall Incorporation ISBN 0-13-6272-58-4. 504p.

Koza, J.R. (1992*). Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press. ISBN 0-262-11170-5. 840 p.

Koza, J.R. (1994*). Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press. ISBN 0-262-11189-6. 768 p.

Koza, J.R.; Bennett, F.H.; Andre, D. & Keane, M.A. (1999). *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann. ISBN 1-55860-543-6. 1154 p.

Koza, J.R.; Keane, M.A.; Streeter, M.J.; Mydlowec, W.; Yu, J.; Lanza, G. (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Springer. ISBN 1-4020-7446-8. 590 p.



Bio-Inspired Computational Algorithms and Their Applications
Edited by Dr. Shangce Gao

ISBN 978-953-51-0214-4
Hard cover, 420 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

Bio-inspired computational algorithms are always hot research topics in artificial intelligence communities. Biology is a bewildering source of inspiration for the design of intelligent artifacts that are capable of efficient and autonomous operation in unknown and changing environments. It is difficult to resist the fascination of creating artifacts that display elements of lifelike intelligence, thus needing techniques for control, optimization, prediction, security, design, and so on. Bio-Inspired Computational Algorithms and Their Applications is a compendium that addresses this need. It integrates contrasting techniques of genetic algorithms, artificial immune systems, particle swarm optimization, and hybrid models to solve many real-world problems. The works presented in this book give insights into the creation of innovative improvements over algorithm performance, potential applications on various practical tasks, and combination of different techniques. The book provides a reference to researchers, practitioners, and students in both artificial intelligence and engineering communities, forming a foundation for the development of the field.