INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2017) Vol. 6 (4) 343−355
DOI: 10.22630/ISIM.2017.6.4.8

EVOLUTIONARY ALGORITHM INSPIRED BY THE METHODS OF QUANTUM COMPUTER SCIENCES FOR THE IMPROVEMENT OF A NEURAL MODEL OF THE ELECTRIC POWER EXCHANGE

JERZY TCHÓRZEWSKI, DARIUSZ RUCIŃSKI

Department of Modelling and Design of Computer Sciences Systems, Institute of Computer Science, Faculty of Sciences, Siedlce University of Natural Sciences and Humanities

The work contains results of research on the possibility to improve the neural model of the Electric Power Exchange (polish: Towarowa Giełda Energii Elektrycznej – TGEE) in MATLAB and Simulink environment using evolutionary algorithm inspired by quantum computer science. The developed artificial neural network was trained using data for the Day Ahead Market, assuming the joint volume of supplied and sold electrical energy [MWh] as the input quantities in each hour of the 24-hour day, and average prices [PLN/MWh] as output quantities. The obtained model of the exchange system was improved using the evolutionary algorithm, and further improvement in the accuracy of the model by supplementing the evolutionary algorithm using quantum solutions, related to the initial population, crossover and mutation operators, selection, etc. were proposed.

Keywords: Artificial Neural Network, Matlab Language, Modelling, Quantum Computation, Polish Power Exchange, Day Ahead Market

1. Introduction

Currently, there are numerous possibilities to model systems and processes in MATLAB and Simulink environment using, i.a. Matlab language and appropriate toolboxes such as, e.g. Neural Network Toolbox (NNT) – neural modelling, Optimization Toolbox (OPT) – evolutionary modelling, Fuzzy Logic Toolbox (FLT) – fuzzy modelling, etc. Special attention was drawn to neural modelling method and evolutionary modelling method. Using the first method, a neural model (NM) of Polish Power Exchange (polish: Towarowa Giełda Energii Elektrycznej (TGEE)) was obtained, and using the second method, the model was improved. As a result, an improved evolutionary-neural model was obtained [13, 17]. Further improvement and refinement of the neural model of the TGEE system using quantum solutions such as, i.a. initial quantum population and quantum crossover and mutation operators, quantum fitness function, quantum selection method was proposed.

In this way, quantum-evolutionary modelling method for the improvement of the parameters of the neural model was obtained. The experiment was based on numerical data related to the TGEE system functioning within the company named Towarowa Giełda Energii S.A. (TGE S.A.), where purchase and sales transactions of electrical energy (ee) as commodity are carried out.

The process of training the artificial neural network (ANN) to learn the model of the TGEE system, numerical data for the Day-Ahead Market (DAM) for the period from 1st January to 30th June 2015 were used. 24 input quantities related to the joint volume of ee in each hour of the 24-hour day [MWh] and 24 output quantities related to the average prices obtained for ee sold [PLN/MWh][1] were assumed. The analysis of the literature on the subject showed that there were no works related to the possibility of using MATLAB and Simulink for the improvement of parameters of neural models of TGEE system using evolutionary method inspired by quantum computer science [1-8, 10, 12, 14].

On the other hand, though, growing demand may be observed for methods of modelling of the TGEE system, including those related to the construction of replacement schemata for the purpose of testing sensitivity, and for simulations including forecast simulations [18, 20]. For the above mentioned reasons, both TGEE system and TGE S.A.[2] as a subject of commercial law are within the scope of interest of both practice and science. Despite numerous attempts including works [1-8, 10, 12, 14], results of research related to the discussed problem, i.e. design and development of a quantum evolutionary algorithm in MATLAB and Simulink environment for the improvement of parameters of the neural model of TGEE within DAM have not been published so far.

[1] so called average price weighted by the volume of all transactions during a stock exchange session for continuous quotations for a given hour of the 24-hour day.

2. Neural modelling of the TGEE system

Neural modelling involves designing the ANN and teaching it the model of the system [11, 15]. At present, due to huge data amounts, the process of neural modelling requires to be preceded by cluster analysis, etc. that involves initial preparation of data for further processing [21]. Matlab language and Neural Network Toolbox (NNT) library were used while designing and teaching the ANN the model of the Polish Power Exchange system (TGEE), which was extensively described, e.g. in works [13, 18, 20, 24]. The following artificial neural network training rule was used in the algorithm:

$$[net, tr] = train(name, X, T, X_i, A_i, EW), \tag{1}$$

which rule returns as the object the artificial neural network defined in variable *net* that consists of, i.a. matrices of weight values, vectors of bias values, etc., where: name – network name, X – input matrix of the network, T – expected values (Target), X_i – initial input lag of the ANN, A_i – initial lag of the hidden layer ΔW – errors related to weight values. Input and output data were normalized in order to improve the process of generalization of learning [21]. As a method of training the ANN, function "*trainlm*" was used, which is an implementation of Levenberg-Marquardt (LM) algorithm, which was extensively described in works [11, 15]. The implementation of the ANN in the MATLAB and Simulink environment using the NNT, allowed for obtaining the neural model of the TGEE system in the component form, composed of two layers, with adder for the first layer determined first, followed by determination of summator for the second layer, according to the dependence:

$$net_j^k = \sum_{i,j} w_{ij}^k \cdot u_i^k, \tag{2}$$

where:

w_{ij} – weight between i-th input to k-th layer of weights and j-th output,

$u_i(t)$ – i-th input to k-th layer of weights at moment t,

net_j^k – sum of products of weights and input quantities to k-th layer as the argument of the activation function.

Next, the activation function was determined, for the first layer tansig() according to the dependence:

$$y_j^1(t) = f(net_j^1) = \frac{2}{1 + e^{-2 \cdot net_j^1}} - 1, \tag{3}$$

and for the second layer according to the dependence:

$$y_j^2(t) = f(net_j^2) = net_j^2, \tag{4}$$

where:

w_{ij} – weight between i-th input to k-th layer of weights and j-th output,

$u_i(t)$ – i-th input to k-th layer of weights at moment t,

net_j^k – sum of products of weights and input quantities to k-th layer as the argument of the activation function,

$y_j^k(t)$ – j-th output from k-th layer at moment t, with k=1 for the output from the first layer and k=2 for the output from the second layer.

Examples of selected values of elements of weight matrix **W1** (with the dimensions 24 x 24 elements) were presented in works [13, 17]. Training the ANN involved modification of values of weights in each layer in order to obtain a model resembling the real system. As a result of training, matrices of weights were equipped with knowledge that, after the completion of the process of ANN training, was in the values of relations between the neurons of the input layer, and the neurons of the hidden layer (**W1**) as well as between neurons of the hidden layer and neurons of the output layer (**W2**), and in biases.

3. Evolutionary modelling of TGEE in MATLAB environment

In order to obtain evolutionary model of ANN, a modified systematic evolutionary algorithm (MSEA) was developed, published, i.a., in works [16-17, 19], in which the modified elements included the method to obtain the initial population (IP) and, consequently, parent populations (PP), as well as the method of obtaining the adaptation function. In order to improve the parameters of the ANN, the MSAE algorithm was implemented in Matlab language [13, 17], which resulted in the improvement of the parameters of the TGEE model as early as in the fourth generation (Fig. 1), and was visible in subsequent generations of the SAE algorithm, with minor changes in the degree of accuracy ranging from 0.9988 to 0.99832, i.e. the average level of adaptation to the environment of 0.9981.

Finally, weight values were obtained with the degree of accuracy from 0.9988 to 0.99832, i.e. at the average level of adaptation to the environment 0.9981 [17]. Selected values were presented in works [13, 17]. In the model, the crossover operator was used, since the mutation operator did not give the desired results, which might have been caused by, i.a. the exhaustion of possibilities of improvement of adaptation during the crossover, and, consequently, further significant improvement was no longer possible.

Therefore, the lack of significant improvement of the neural model as a result of the mutation operator was caused by very high degree of adaptation of the neural model to the TGEE system equal to 0.9981. Thus, other methods, such as quantum computer science methods were required to obtain further improvement of the neural model [3, 13, 16-17].

In order to improve the parameters of the obtained neural model of the TGEE (NM), in the considered case the values of both weight matrices **W1** and **W2,** a neural-evolutionary model (NEM) was developed, which allowed to improve the accuracy of the TGEE model in relation to the real model.

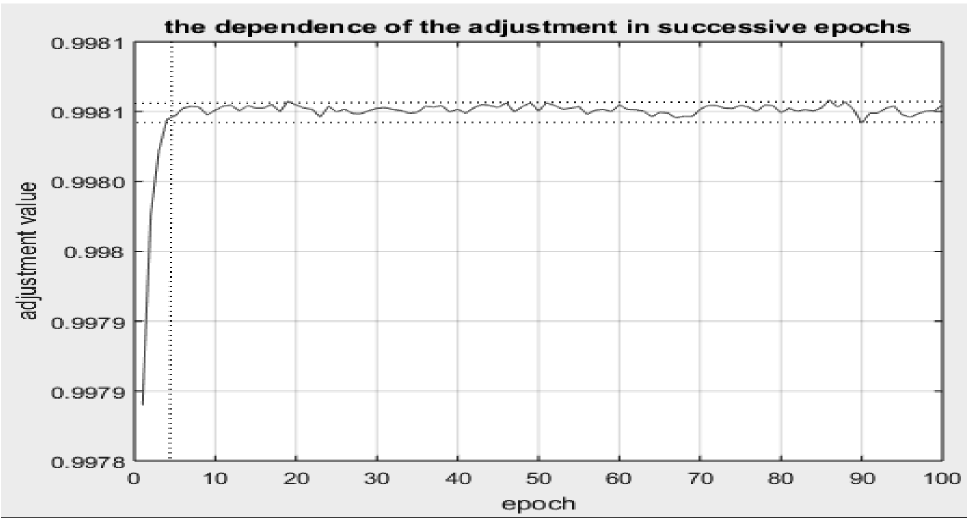

Figure 1. The graph of adaptation (robustness) function relative to epoch in EA. Denotations: epoch – number of subsequent population, value of adaptation – the ratio between the reference value to the value obtained from the model. *Source*: [17]

Results obtained from the neural model of the TGEE and neural-evolutionary model (neural model improved by means of SEA algorithm, based on Systematic Evolutionary Algorithm) were compared using a simulation model developed in Simulink [13, 17], which allowed to obtain the courses of selected output quantities, as in Fig. 3. The analysis of the discrepancies shows that the output quantities generated by the NEM were closer to the real system than the quantities generated by the NM [13, 17].

4. Quantum Evolutionary Algorithm for the improvement of parameters of the neural model

In order to obtain the evolutionary model, a modified method of obtaining the adaptation function (AF) was developed. A basic block diagram was assumed, related to the modified systematic evolutionary algorithm (SEA) [18, 19-21], in which the following were determined using quantum methods: initial population (IP), robustness (adaptation) function (AF), crossover operator (CO) and mutation operator (MO). Additionally, selection method appropriate for the conditions of the problem being solved was chosen.

Next, the modified SEA algorithm, supplemented with quantum calculations (QEA) was used to improve the parameters of the neural model of the TGEE. For that purpose, quantum calculations were performed, which required performing the quantization of decimal numbers to quantum numbers, succeeded by dequantization of quantum numbers to decimal numbers when the calculations were completed. Both the process of quantization and dequantization requires the adoption of certain methods of conversion of decimal numbers to binary numbers, and, further, to quantum numbers [16-17].

For the purpose of decimal numbers quantization, it is convenient to convert, e.g. the values of parameters of the neural model of TGEE, from decimal numbers into binary numbers, and assume that classical bits 0 and 1 are clean states, with the assumption that the quantum number obtained from the binary number represents clean states. The problem of computational complexity appears when mixed states of a quantum number are determined, even for one-qubit register. Then, it is possible to determine any number of mixed states for a given quantum number, with the assumption of the occurrence of superposition principle. It is then important to determine the probability of occurrence of |0> and |1> based on the superposition principle, i.e. after the assumption of boundary state described by the equilibrium state, e.g. assuming that $\alpha = \beta$.

Hence, from the superposition principle described by the following dependence: $2\alpha^2=1$ the boundary between the areas of the two clean states with the value $\alpha=\beta= 0.71$ is obtained. In order to obtain mixed states, the following steps are performed: for a clean state |0> (if it dominates) – a value from the interval (0.71÷1) is selected at random, and for a clean state |1> (if it dominates) – a value from the interval (0.71÷1) is selected at random, i.e. for the value of the clean state |0> a mixed state |M0> with the domination of clean state |0> is obtained, i.e. random selection from the dominant interval $0.71 \leq \alpha \leq 1$ and random selection of β from recessive interval $0 \leq \beta \leq 0,71$, and similarly, for values of clean state |1> a mixed state ought to be obtained |M1> with the domination of clean state |1>, i.e. random selection of β from the dominant interval $0.71 \leq \beta \leq 1$ and random selection of α from recessive interval $0 \leq \alpha \leq 0.71$.

The above mentioned method allows for specification of unequivocal yet random method of determination of mixed states using a classical computer. Further, it is worth noticing that e.g. due to the fact that there may occur an infinite number of mixed states (let us assume that in this case, the infinity is the number of computational data e.g. 1000) in each qubit (which is a 2-row matrix), the number of occurrences |0> in 1000 cases for a clean state |0> is then selected at random, and the number of occurrences |1> is calculated based on the superposition principle, and, similarly, the number of occurrences |1> in 1000 cases for a clean state |1> , and the number of occurrences 0> is calculated based on the superposition principle $(\alpha^2+\beta^2=1)$. In this way, pairs of mixed numbers are obtained, which are then further used for the purpose of narrowing the areas of occurrence of mixed states of quantum numbers.

5. Possibilities of Improvement of the ANN using the Quantum Evolutionary Algorithm

The possibility of improvement of ANN parameters, including the weights in both layers arises from increasing the degree of accuracy of the problem being solved, in accordance with the higher precision inspired by the quantum mechanics and computer science.

One of first attempts to implement evolutionary algorithms inspired by quantum computer science was the work by Naranyan & More, 1996 [10], in which an attempt to implement a new EA was made, based on quantum calculations logic, and the conception of the existence of so called parallel universes. The proposed method was verified using the example of so called travelling salesman problem. Each universe was represented by its own population of chromosomes, and each chromosome was represented by a two-dimensional matrix with one row representing the route, and a letter representing cities, with the assumption that every route determined in the above mentioned way was different. Crossover involved the determination of a new generation of chromosomes by selecting individual genes from different universes, and if the letter repeated, a jump to the next different generation occurred. For this model, a proof proving that quasi-quantum approach used in the evolutionary algorithm is better than the classic evolutionary algorithm was not given.

Also, a method of representation of individual chromosomes using quantum gates was not indicated, i.e. neither a method for quantum representation of chromosomes nor alternative universes by means of quantum-evolutionary approach was not shown. Nevertheless, the accuracy of the travelling salesman problem being solved was improved. Another work, which presents results related to the implementation of quantum evolutionary algorithm, using quantum computer, was work by K. Han & J. Kim, 2000 [2], and shows the method of quantum-evolutionary solution using the example of the discrete backpack problem.

It was assumed that qubits represent base states of the chromosome, and a register is used for the representation of superposition of states of individual qubits. The process of improvement of the parameters of the model was implemented using rotation gates. This method used neither crossover algorithm nor mutation algorithm, which, obviously, limited the possibilities of finding a robust solution. The problem related to the implementation of the above mentioned algorithm was the lack of possibility to measure the quantum state, which, changed at the moment of recording, which, in practice, made it impossible to use it to measure current quantum states. Therefore, indirect methods were used to perform measurements.

The literature on the subject provides numerous similar attempts at finding quantum-evolutionary algorithms for solving various computational problems, such as works [1, 4-8, 10, 12, 14, 16], related to evolutionary solutions inspired by quantum computer science. However, among them there are no proposals, which would solve the problem of quantum calculations, including obtaining and using mixed numbers in the calculations, in the appropriate way, and increase the accuracy of modelling. The problem of mixed numbers was described, i.a. in works [16], in which a method of obtaining quantum numbers was proposed. Based on the above mentioned method of obtaining quantum numbers and fundamentals of quantum evolutionary algorithm, a method of quantum calculations that uses classical computers was developed. The method comprises the basic steps presented on Fig. 4. Let the need to determine quantum value of the adder net with the structure as in Fig. 2 arise. Determination of quantum numbers is shown on Fig. 3, using next equation:

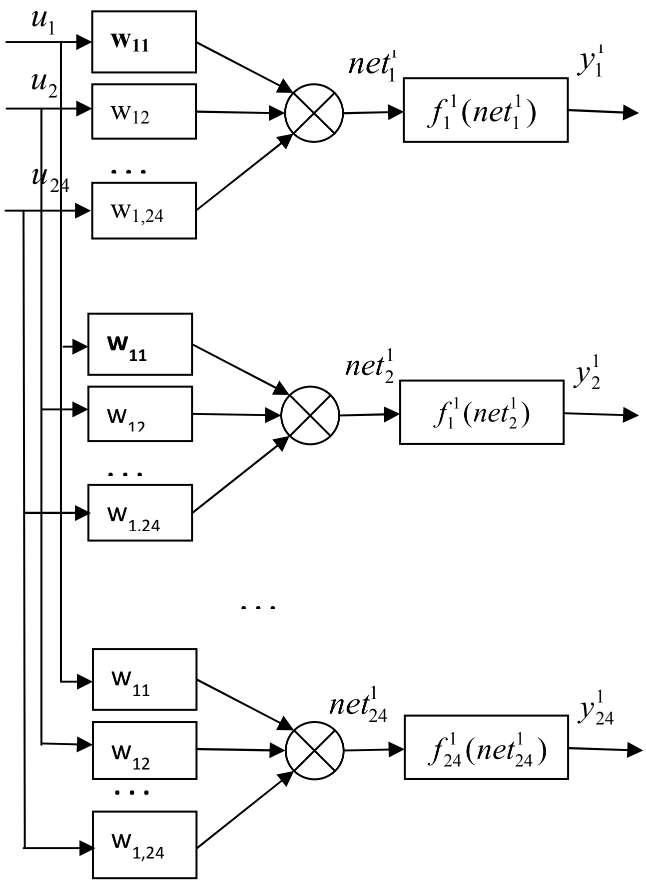

Figure 2. Adder net_1. Denotations: $u_1 \ldots u_{24}$ – input signals to the ANN (here: the volume of ee in each hour of the 24-hour day), y_1 – output signals from the ANN (here: average selling price of ee), $w_{11}, w_{12} \ldots w_{1,24}$ – weights of layer **W1** connected to neuron 1 (net_1^1).

Step 2. Conversion of values written in the binary system for input u_1 and weight $w_{1,1}$ here: 01 January 2015) into values in the quantum system (the dimension of state vector in Hilbert space must be the same as the dimension of weights vector): decimal values: binary values: u_1=[0.000010001101], $w_{1,1}$=[0.010011100001], quantum values (to two digits after the point):

$$u_1 = \begin{bmatrix} 0.87 & 1 & 0.77 & 0.98 & 0.67 & 0.74 & 0.89 & 0.85 & 0.14 & 0.37 & 0.77 & 0.44 \\ 0.49 & 0 & 0.64 & 0.20 & 0.74 & 0.67 & 0.46 & 0.53 & 0.99 & 0.93 & 0.64 & 0.90 \end{bmatrix},$$

$$w_{1,1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.51 & 0.49 & 0.52 & 0.40 & 0.38 & 0.49 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.49 & 0.51 & 0.48 & 0.60 & 0.62 & 0.51 \end{bmatrix}. \quad (5)$$

$$net_1^1(u) = w_{11}^1 \cdot u_1 + w_{21}^1 \cdot u_2 + \ldots + w_{24,1}^1 \cdot u_{24} = \begin{bmatrix} 10.6904 & 6.0230 \\ 9.5368 & 5.6529 \end{bmatrix}. \quad (6)$$

$$net_1^1(u) = \begin{bmatrix} net_{11}^1 & net_{12}^1 \\ net_{21}^1 & net_{22}^1 \end{bmatrix} = \begin{bmatrix} \alpha_1^1 \cdot \alpha_2^1 & \beta_2^1 \cdot \alpha_1^1 \\ \beta_1^1 \cdot \alpha_2^1 & \beta_1^1 \cdot \beta_2^1 \end{bmatrix} = \begin{bmatrix} 0.6904 & 0.0230 \\ 0.5368 & 0.6529 \end{bmatrix}. \quad (7)$$

with, e.g.:

$$net_{11}^1(u_1) = w_{11}^1 \cdot u_1 = \left(\frac{0.87}{2^0} \cdot \frac{0}{2^0}\right) + \left(\frac{1}{2^1} \cdot \frac{1}{2^1}\right) + \ldots + \left(\frac{0.44}{2^{11}} \cdot \frac{0.49}{2^{11}}\right) = \left(\frac{1}{2^1} \cdot \frac{1}{2^1}\right) + \left(\frac{0.67}{2^4} \cdot \frac{1}{2^4}\right) + \ldots + \left(\frac{0.44}{2^{11}} \cdot \frac{0.49}{2^{11}}\right) =$$
$$== \left(\frac{1}{2^2}\right) + \left(\frac{0.67}{2^8}\right) + \ldots + \left(\frac{0.44 \cdot 0.49}{2^{22}}\right) = \left(\frac{1}{2^2}\right) + \left(\frac{0.67}{2^8}\right) + \ldots + \left(\frac{0.2156}{2^{22}}\right) = 0.25279485$$

$\alpha_1\alpha_2$= 0.6902, 2) $\alpha_1\beta_2$= 0.0230, 3) $\beta_1\alpha_2$=0.5328, 4) $\beta_1\beta_2$=0.6529, respectively. (8)

$$(\alpha_2)^2 + (\beta_2)^2 = 1. \quad (9)$$

$$\alpha_1 = \frac{0.6902}{\alpha_2}, \quad (10)$$

$$\alpha_1 = \frac{0.0230}{\beta_2}, \quad (11)$$

$$\alpha_2 = 30.01 \cdot \beta_2. \tag{12}$$

$$(30.01 \cdot \beta_2)^2 + {\beta_2}^2 = 1, \tag{13}$$

$$\beta_2 = 0.03330374 \text{ and } \alpha_2 = 0.999944528. \tag{14}$$

α_1=0,6902/α_2=0,69058308, β_1=0,5368/0.99944528=0,53709794, α'_1=0.68036906.

$$net_1^1(u) = \begin{bmatrix} net_{11}^1 & net_{12}^1 \\ net_{21}^1 & net_{22}^1 \end{bmatrix} = \begin{bmatrix} \alpha_1^1 \cdot \alpha_2^1 & \beta_2^1 \cdot \alpha_1^1 \\ \beta_1^1 \cdot \alpha_2^1 & \beta_1^1 \cdot \beta_2^1 \end{bmatrix} = \begin{bmatrix} 0{,}69058308 \cdot 0{,}99944528 & 0{,}69058308 \cdot 0{,}03330374 \\ 0{,}99944528 \cdot 0.7225307 & 0.7225307 \cdot 0{,}03330374 \end{bmatrix} =$$
$$= \begin{bmatrix} \alpha_1^1 \\ \beta_1^1 \end{bmatrix} \cdot \begin{bmatrix} [\alpha_2^1 & \beta_2^1] \\ [\alpha_2^1 & \beta_2^1] \end{bmatrix} = \begin{bmatrix} 0{,}69058308 \\ 0.7225307 \end{bmatrix} \cdot \begin{bmatrix} [0{,}99944528 & 0{,}03330374] \\ [0{,}99944528 & 0{,}03330374] \end{bmatrix} = l_{m1}^1 \cdot \begin{bmatrix} l_{m2}^{1T} \\ l_{m2}^{1T} \end{bmatrix} =$$
$$= l_{m1}^1 \cdot l_{m2}^{1T} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = l_{m1}^1 \cdot l_{m2}^{1T} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = l_{m1}^1 \cdot l_{m2}^{1T} \cdot |1\rangle \cdot |0\rangle. \tag{15-1}$$

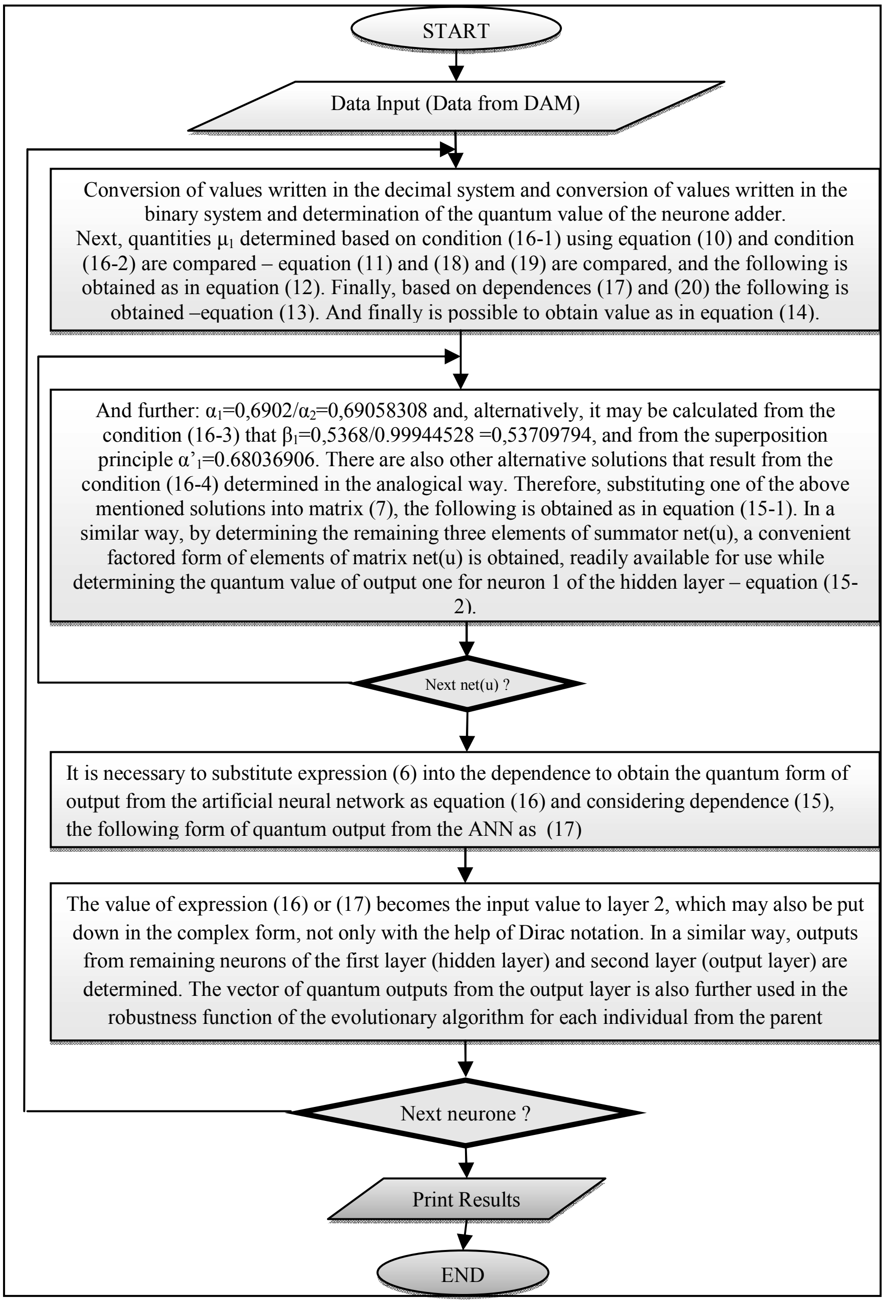

Figure 4. Determination of quantum numbers

$$net_{12}^{1}(u) = l_{12m1}^{1} \cdot l_{12m2}^{1T} \cdot |1\rangle \cdot |0\rangle.$$

$$net_{21}^{1}(u) = l_{21m1}^{1} \cdot l_{21m2}^{1T} \cdot |1\rangle \cdot |0\rangle. \tag{15-2}$$

$$net_{22}^{1}(u) = l_{22m1}^{1} \cdot l_{22m2}^{1T} \cdot |1\rangle \cdot |0\rangle.$$

$$y_1^1(t) = f(net_1^1) = \frac{2}{1+e^{-2\cdot\begin{bmatrix}10.6904 & 6,0230\\ 9.5308 & 5,6529\end{bmatrix}}} - 1 = \frac{2}{1+e^{-\begin{bmatrix}21 & 12\\ 11 & 11\end{bmatrix}} \cdot e^{-\begin{bmatrix}0.3808 & 0.0460\\ 0.0736 & 0.3058\end{bmatrix}}} - 1, \tag{16}$$

$$y_1^1(t) = f(net_1^1) = \frac{2}{1+e^{-2\begin{bmatrix}net_{11}^1 & net_{12}^1\\ net_{21}^1 & net_{22}^1\end{bmatrix}}} - 1 = \frac{2}{1+e^{-2\begin{bmatrix}l_{11m1}^1 \cdot l_{11m2}^{1T} \cdot |1\rangle \cdot |0\rangle & l_{12m1}^1 \cdot l_{12m2}^{1T} \cdot |1\rangle \cdot |0\rangle\\ l_{21m1}^1 \cdot l_{21m2}^{1T} \cdot |1\rangle \cdot |0\rangle & l_{22m1}^1 \cdot l_{22m2}^{1T} \cdot |1\rangle \cdot |0\rangle\end{bmatrix}}} \tag{17}$$

6. Conclusions

Quantum evolutionary algorithm was proposed, which was verified using numerical examples for the problem related to the improvement of parameters of the neural-evolutionary model of TGEE. For the purpose of quantum-evolutionary algorithm development, theory related to quantum computer science, and in particular, author's own method of mixed quantum numbers determination was applied [16-17]. The research is continued in order to determine the degree of improvement of parameters of the neural-evolutionary model by supplementing the modified SEA algorithm with solutions from the field of quantum computer science.

REFERENCES

[1] Araujo R. de A., Aranildo Junior R.L.; Tiago Ferreira A. E. Tiago (2008) *A Quantum-Inspired Intelligent Hybrid method for stock market forecasting*, 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp.1348–1355.

[2] Han K., Kim J. (2000) *Genetic quantum algorithm and its application to combinatorial optimization problem*, Evolutionary Computation, Proceedings of the 2000 Congress on, Vol. 2, pp. 1354 – 1360.

[3] Hirvensalo M. (2004) Quantum algorithm, WSiP, Warszawa 2004.

[4] Ho S.L., Yang S., Ni P., Huang J. (2013) *A Quantum-Inspired Evolutionary Algorithm for Multi-Objective Design*, IEEE Transactions on Magnetics, Volume: 49, Issue 5, pp.1609 – 1612.

[5] Kasabov N. (2006) *Neuro-, Genetic-, and Quantum Inspired Evolving Intelligent Systems*, International Symposium on Evolving Fuzzy Systems, pp. 63 – 73.

[6] Li J., Li J. (2008) *Next-Day Electricity Price Forecasting Based on Support Vector Machines and Data Mining Technology*. Proceedings of the 27th Chinese Control Conference, Kunming, Yunnan, China, pp. 630 – 633.

[7] Liao G. (2010) *Using chaotic quantum genetic algorithm solving environmental economic dispatch of Smart Microgrid containing distributed generation system problems*, Power System Technology, POWERCON, International Conference on, pp. 1 – 7.

[8] Liu Y.Y., Li M. (2008) *A bidding method of power market based on immune genetic algorithm*, 27th Chinese Control Conference, pp. 28 – 32.

[9] Mielczarski W. (2000) *Rynki energii elektrycznej. Wybrane aspekty techniczne i ekonomiczne*, ARE, Warszawa (in Polish).

[10] Narayanan A., Moore M. (1996) *Quantum-inspired genetic algorithms*, Proceedings of IEEE International Conference on Evolutionary Computation, pp. 61-66.

[11] Osowski S. (2000) *Sieci neuronowe do przetwarzania informacji*, PW, Warszawa (in Polish).

[12] Pittenger A.O. (2000) *An Introduction to Quantum Computing Algorithms*, Birkhauser, Boston.

[13] Ruciński D. (2016) *Neural-evolutionary Modelling of Polish Electricity Power Exchange*, XPlore Digital Library, EPNet, PWr.

[14] Sheng Z., J. Wanlu (2012) *A novel quantum genetic algorithm and its application. Natural Computation (ICNC)*, Eighth International Conference on, pp. 613 – 617.

[15] Tadeusiewicz R., Szaleniec M. (2015) *Leksykon sieci neuronowych*, Wyd. Fundacji „Projekt Nauka”, Wrocław (in Polish).

[16] Tchórzewski J., Wołynka Ł. (2016) *Możliwości informatyki kwantowej do poprawy dokładności modelowania. Część 2. Kwantowy algorytm ewolucyjny ruchu robota PR-02*. Poznan University of Technology. Academic Journal. Wyd. PP, Poznań (in Polish).

[17] Tchórzewski J., Ruciński D. (2016) *Quantum Inspired Evolutionary Algorithm to Improve Parameters of Neural Models on Example of Polish Electricity Power Exchange. EPNet. Xplore Digital Library.* International Conference on Electrical Power Networks EPNet. PWr, Wrocław-Szklarska Poręba.

[18] Tchórzewski J. (2013) *Rozwój systemu elektroenergetycznego w ujęciu teorii sterowania i systemów*. OW PWr. Wrocław (in Polish).

[19] Tchórzewski J. (2005) *Systemowy Algorytm Ewolucyjny SAE*, Bio-Algorithms and Med-Systems, Journal Edited by Collegium Medicum, Vol. 1. No. ½, UJ, Kraków, pp. 61-64.

[20] Tchórzewski J. (1990) Inżynieria rozwoju systemów. Monografie nr 18, Wydawnictwa Uczelniane WSR-P w Siedlcach, Siedlce, Poland (in Polish).

[21] Wierzchoń S., Kłopotek M. (2015) Cluster analysis. Monografie. Wyd. IPI PAN. Warszawa.