Computer Applications in Electrical Engineering
Vol. 14 *2016*
DOI 10.21008/j.1508-4248.2016.0025

Systemic Evolutionary Algorithm inspired by methods of Quantum computer sciences for the improvement of the accuracy of neural models in electrical engineering and electrical power engineering

Jerzy Tchórzewski
Siedlce University of Natural Sciences and Humanities
08–110 Siedlce, 3-Maja 54, e–mail: Jerzy.Tchorzewski@uph.edu.pl

The work contains selected results of research on the application of quantum computer science to a systemic evolutionary algorithm for the purpose of improving accuracy of neural models in electrical engineering and electrical power engineering. Artificial neural networks are used in neural modeling, which networks are designed and taught models of systems using available numerical data. Parameters of neural networks, and especially, elements of weight matrices, biases as well as parameters of activation functions may be improved using evolutionary algorithms. It seems that applying solutions offered by quantum computer science to systemic evolutionary algorithm, and especially, as regards creation of quantum initial population, quantum crossover and mutation operators as well as selection, considerably improves the accuracy of modelling, which was verified in MATLAB and Simulink environment using selected examples such as RP–02 robot's arm movement, the development of the Polish Electrical Power Exchange (polish: TGEE) system, etc.

KEYWORDS: Evolutionary Algorithms, Quantum computer science, Quantum mixed number, Systems modelling, Robot PR–02, Artificial Neural Networks, MATLAB and Simulink environment, Electric Power Exchange

1. The possibility of quantum computing in modeling systems

1.1. Systemic Evolutionary Algorithm

At present, one of primary challenges in applied electrical engineering is the problem of finding more accurate methods of modeling inspired by increasing miniaturization of circuits and systems of control [3, 7, 13, 29, 49–50]. Among promising methods used to improve parameters of models of systems in electrical engineering and electrical power engineering, apart from identification and neural models [15, 25, 30–31, 37, 51–52], there are evolutionary modeling methods such as genetic algorithms (GA) and evolutionary algorithms (EA) [10, 17, 19, 23, 34–35, 38, 40–41, 44–45, 51].

Due to the fact that better accuracy of systems models parameters may be obtained by combining evolutionary methods and methods inspired by quantum computer science [1–2, 9, 14, 18, 20–22, 24, 26, 32–33, 36, 39, 42–43], an attempt was made at developing Quantum Evolutionary Algorithm (QEA) [32–33, 38, 40, 42–43] based on the original Systemic Evolutionary Algorithm (SEA) [34, 36, 39, 41, 44–45] using systems control and development theory [16, 37, 46–48].

There are numerous approaches to evolutionary and genetic algorithms, i.a., methods based on classic genetic algorithm, evolutionary methods and strategies, co–evolutionary methods, etc. [10, 17, 19, 23, 44]. The main unsolved problem in genetic and evolutionary algorithms remains appropriate generation of the Initial Population (IP), and resulting Parent Populations (PP), as well as the problem of constructing Robustness Function (RF), often wrongly brought down in literature of the subject to adaptation function on the grounds of optimization methods [38, 40–41, 44–45].

In SEA algorithm, presented in work *Systemic Evolutionary Algorithm* in Journal edited by Collegium Medicum entitled *Bio–Algorithms and Med–Systems* [44], generation of the initial population, as the information about the structure and parameters of the system model as well as Robustness Function was defined in a systemic way, using the notion of systemic discrepancy occurring between the average value of the population and the value of each individual in each generation of the SEA [7, 13, 37, 46–48]. Basic operators crossover, mutation and selection were defined on the grounds of systems control and development theory [16, 37, 46–48]. SAE algorithm was developed and implemented using methods inspired by quantum computer science, which allowed to obtain QEA [38, 40–41, 45].

Verification involving experiments was conducted using practical examples such as PR–02 robot's arm movement, development of TGEE, development of electrical power system, etc. [32–33, 36, 39, 42–43] implemented in MATLAB and Simulink environment.

1.2. Potential of quantum computer science

Quantum computer science is considerably new and little known field of science using Hilbert's space, where enormous amounts of information may be stored, not comparable with classic optic storage devices [6, 8, 11–12, 27, 29, 49]. One of basic terms connected with quantum computer science is state of object in quantum sense, e.g. state of molecule described by its wave function, which cannot be observed in the case of classic objects, but is processed and passed between quantum systems in nature, which is not perceived by man – a classic (deterministic) observer – [11–12, 27, 29].

Elementary medium of quantum information is a qubit, also called a quantum bit. A quantum bit has two pure states and an infinite number of mixed states (intermediate values). Quantum information is described using quantum numbers. State of a quantum number may be described in a two–dimensional complex Hilbert's space with an ortonormal base {|0>, |1>} using vector [3, 6, 8, 27, 29, 49]:

$$|l_k\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where numbers α and β are complex numbers (so called modules of probability) that fulfil for qubit the condition of superposition:

$$\alpha^2 + \beta^2 = 1, \tag{2}$$

defining them as amplitudes of probability, with a certain state, in which any qubit as a linear combination of vectors from a defined base [27, 29, 49]. In vector notation, in accordance with Dirac notation, qubit is presented as a ket in the form of:

$$|l_k\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{3}$$

where for qubit |0> the value $\alpha = 1, \beta = 0$, and for qubit |1> value $\alpha = 0, \beta = 1$, thus pure states for qubit are unequivocally specified. Therefore, a quantum bit may be in a superposition of two base states, and while processing quantum state in a superposition, all possible combinations of intermediate states are also taken into consideration. E.g. for one qubit, whose base are two pure states, their superposition that operates on both those states in the same time, determines an infinite number of intermediate states. It means that it is impossible to unequivocally specify the value of mixed state, but it is possible to unequivocally speak about the module of probability, with which a mixed state may be observed (pure states with probability amplitudes α and β). Superposition occurs when pure states do not occur ($\alpha, \beta \neq 0$) [12, 27, 29]. Determination of the method to obtain mixed states using classic computers is connected with developing appropriate method using quantum function. Similarly for a qubit[1], i.e. for a unit of quantum information, resulting from a generalized notion of qubit, in which, d base states occur, i.e. d states are in superposition state, which may be written after work *Informatyka kwantowa. Wybrane obwody i algorytmy [Quantum computer science. Selected circuits and algorithms]* by M. Sawerwain and J. Wiśniewska [27] in the following way:

$$|\Psi\rangle = \sum_{i=1}^{d} \alpha_i \cdot |i\rangle \quad \textbf{\textit{and}} \quad \sum_{i=1}^{d} |\alpha_i\rangle^2 = 1, \tag{4}$$

[1] In English–language literature, qubit is defined as freedom level (degree), after work [27].

where: α_i – i–th base state of a qubit, |i> – computational base for a given quantum register, with the standard base ranging from 0 to 2^n–1, d – qubit degree, i.e. the number of base vectors.

It is worth noting after work by K. Giaro and M. Kamiński that more than one qubit is involved quantum computations. Thus, a notion of qudit register is defined, i.e. Hilbert's space vector H_2^n with unit length, i.e. the tensor product n of the Hilbert's space H_2 [6]. Quantum register that contains n qubits is obtained using the operation of tensor product in the following form [27]:

$$|\Psi\rangle = |\Psi_0\rangle \otimes |\Psi_1\rangle \otimes \ldots \otimes |\Psi_{n-1}\rangle, \tag{5}$$

and the state of unknown quantum register may be written:

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i \cdot |i\rangle = \alpha_0 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ \ldots \\ 0 \end{bmatrix} + \alpha_1 \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ \ldots \\ 0 \end{bmatrix} + \alpha_2 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ \ldots \\ 0 \end{bmatrix} + \ldots + \alpha_{2^n-1} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ \ldots \\ 1 \end{bmatrix}, \tag{6}$$

if the normalization condition (superposition of states) is fulfilled, and the rule of marking individual qubits is introduced, e.g. establishing that the first qubit with the index equal zero is positioned on the left of the register, and the last qubit on the right of the quantum register. Three types of operations may be performed on the quantum register, namely, unitary operation (so called unitary evolution), measurement operations and general operations. Unitary operation involves multiplication of the initial state of the qubit s(0) by the unitary matrix U_t in order to obtain qubit state at moment t, i.e.:

$$s(t) = U_t \cdot s(0). \tag{7}$$

Quantum measurement involves applying a set of measurement operators $\{M_i\}$ to a qubit s(t) in order to obtain the result of measurement operation of I type, and the probability of obtaining the result i is equal:

$$p(i) = \langle \Psi | \cdot M_i^+ M_i \cdot |\Psi\rangle, \tag{8}$$

with the state of the system being equal to Ψ prior to performing the measurement, and following the measurement, using the set of operators $\{M_i\}$, it is equal:

$$\Psi(t) = \frac{M_i \cdot |\Psi\rangle}{\sqrt{\langle \Psi | \cdot M_i^+ \cdot M_i \cdot |\Psi\rangle}}. \tag{9}$$

General quantum operations (quantum channels) result from considering the impact of the environment on quantum register, which is connected with the course of unitary evolution, i.e. irreversible revolution. In such a situation, basic part of the system (system S^2) is separated from the additional part of the system

[2] System.

(environment E[3]), which leads to the description of time evolution of the system in the following way:

$$\rho_{SE}^{2} = U \cdot (\rho_{S}^{1} \otimes |e_0\rangle\langle e_0|) \cdot U^{+}, \tag{10}$$

where $|e_0\rangle$ is the initial state of the environment.

1.3. Attempts at implementation of EA inspired by methods of quantum computer science

One of the first attempts at implementation of Evolutionary Algorithms (EA) inspired by methods of quantum computer science was a work by Naranyan & More published in 1996 [15], in which a new EA was implemented based on quantum computations logic, based on the concept of existence, so called parallel universes (worlds). The proposed method was verified based on the example of the solution of the so called travelling salesman problem. Each universe was represented by its own population of chromosomes, and each chromosome by two–dimensional matrix, with one row representing a route and a letter representing cities, with the assumption that no route defined in the way described above repeats.

Crossover involved determination of a new generation of chromosomes by selecting individual genes from different universes, and if the letter repeated a jump to the next not–repeating generation was made. It was not proved for this model that quasi–quantum approach used in EA is better than the classic evolutionary algorithm, and a method of representation of individual chromosomes, and thereby alternative universes using quantum–evolutionary approach, using quantum gates was not indicated, with obtaining improvement in accuracy of the travelling salesman problem being solved.

Another work, in which results related to the implementation of QEA using a quantum computer, was a work by K. Han & J. Kim published in 2000 [6], in which a quantum–evolutionary solution was shown using so called discrete packpack problem as an example. It was assumed that qubits represent base states of a chromosome, and the register represents superposition of states of individual qubits. The process of the model's parameters refinement was implemented using rotation gates. Neither crossover algorithm nor mutation algorithm was used in this method, which, obviously, limited the possibilities of finding a robust solution. A problem connected with the implementation of the above mentioned algorithm was lack of possibility to measure the quantum state that immediately changed at the time of reading, which made it impossible to use it for reliable measurements of current quantum states. Therefore, indirect methods were used to for the purpose of measurements.

[3] Environment.

In literature of the subject, numerous similar attempts to find quantum–evolutionary algorithms for solving various computational problems can be found, e.g. works [1, 5, 9–12, 22] concerning evolutionary solutions inspired by quantum computer science. However, they all lack a solution that would improve the accuracy of modelling and solve the problem of quantum computations, especially, as regards obtaining and using mixed numbers in computations, in the appropriate way. They also include works published by the author [32, 36].

2. Quantum Evolutionary Algorithm

2.1. The Essence of QEA

Quantum evolutionary algorithm is best constructed based on the SEA algorithm, by supplementing the methods used with methods inspired by quantum computer science. A basic flowchart for Systemic Evolutionary Algorithm was adopted, in which the initial population, robustness function as well as crossover, mutation and selection methods were defined based on quantum computer science (Fig. 1) [[38, 41, 44]. As a result, quantum calculations were also conducted using quantum computer science, which required specific quantization of numbers written in a decimal system, i.e. conversion of decimal numbers into quantum numbers. For obvious reasons, quantum numbers were converted back into binary numbers and then to decimal numbers, when calculations were completed. Thus, both quantization and dequantization processes requires choosing a certain method of conduct for the conversion of a binary number into a quantum number and vice versa.

2.2. Quantum Initial Population

An introductory issue related to the SEA algorithm is an issue of analytic, neural or identification modelling for the purpose of obtaining a model of any system. The obtained model has a certain structure and parameters of the highest attainable accuracy. It is convenient to change the values of e.g. model parameters, let the model be a model of robot PR–02 [28, 33], in the form of decimal numbers into binary numbers, and assume that bits 0 and 1 are pure states, with the assumption that a quantum number obtained from a binary number represents pure states. The problem of computational complexity appears when mixed states of a quantum number are determined, even using one–cubit register. Quantum numbers may then be expressed using dependence (1), with the assumption that superposition occurs. It is important to determine the module of probability of occurrence of $|0\rangle$ and $|1\rangle$ based on the rule of superposition, i.e. with the assumption of critical state described by the

equilibrium state, e.g. assuming that $\alpha = \beta$. Hence, from the superposition rule described by interdependence (2) a limit between two areas of both pure states is obtained with the value $\alpha = \beta = 0.71$.

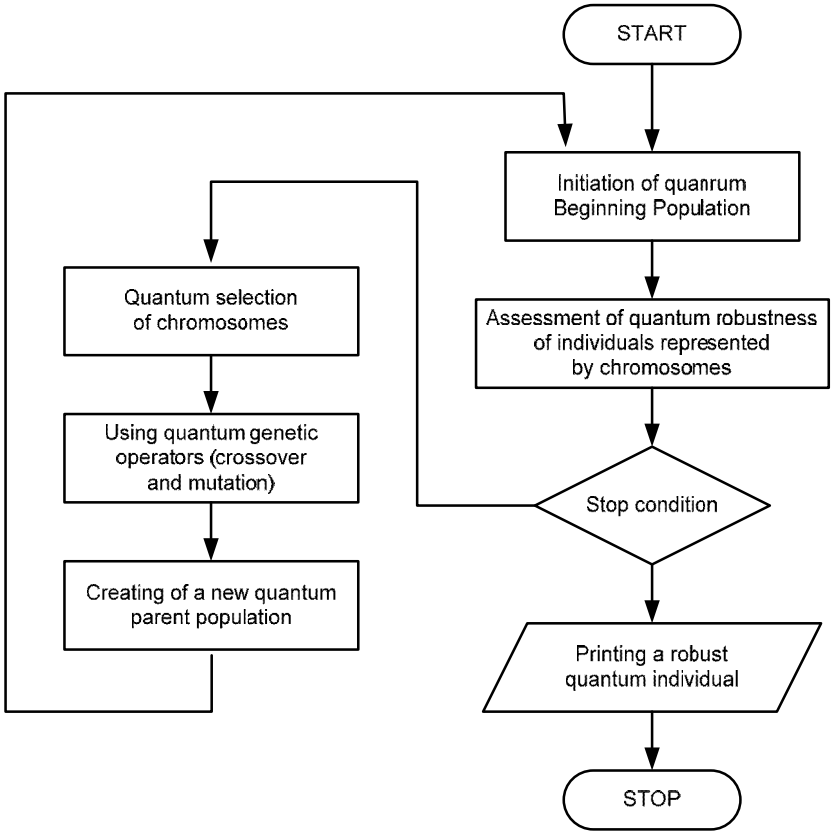

Fig. 1. A general flowchart of QEA. Source: Own elaboration based on [37, 44]

To determine mixed states, it is necessary to perform sampling from the interval (0.71–1) for the pure state |1> z, and from the same interval (0.71–1) for the pure state |0>, thus obtaining a mixed state with the domination of pure state |0> for the value of pure state equal |0>, i.e. sampling from α from the interval $0.71 \leq \alpha \leq 1$ and sampling β from the interval $0 \leq \beta \leq 0.71$. Similarly, for the value of the pure state equal |1>, a mixed state |M1> ought to be obtained with the domination of |1>, i.e. β is sampled from the interval $0.71 \leq \beta \leq 1$ and α is sampled from the interval $0 \leq \alpha \leq 0.71$. In the above mentioned way, an unequivocal, though random method of determining mixed states for classic computer.

Further, it is worth noticing that due to the fact that, an infinite number of mixed states may occur in each qubit (a register that is a matrix), (let us assume that in this case infinity is limited by the amount of computational data e.g. 1,000), the number of occurrences of |0> in 1 000 cases is sampled for a pure state |0>, and the number of occurrences |1> is calculated based on the rule of superposition. Similarly, the number of occurrences |1> in 1 000 cases is sampled for a pure state |1> and a number of occurrences 0> is calculated based on the rule of superposition ($\alpha^2 + \beta^2 = 1$). In this way, pairs of mixed numbers are obtained, which are further used for narrowing areas of occurrence of mixed states of quantum numbers.

2.3. An example of determination of a mixed state

Let us sample a number of occurrences |0> in 1 000 states of a quantum number from the interval of values 0.71 <= α <= 1, obtaining e.g. α = 0.8, e.g. for a pure state |0>, followed by determination of β based on superposition rule, and simultaneously, a number of occurrences of state |1> in 1 000 states of quantum number, but from the interval of values 0 <= beta <= 0.71, obtaining e.g. $\beta = 0.2$, and α is calculated based on superposition rule for pure state |1>.

Thus, the algorithm for determination of mixed states is as follows:

I variant:

Let $\alpha = 0.8$, based on superposition rule β is determined in the following way:

$$(0.8)^2 + \beta^2 = 1, \tag{11}$$

thus $\beta = 0.6$, and a mixed quantum number is equal:

$$N_{qm1} = 0.8 \cdot |0\rangle + 0.6 \cdot |1\rangle. \tag{12}$$

II variant:

Let $\beta = 0.2$, based on superposition rule α is determined:

$$(\alpha)^2 + (0.2)^2 = 1, \tag{13}$$

thus $\alpha = 0.98$, and a mixed quantum number is equal:

$$N_{qm1} = 0.98 \cdot |0\rangle + 0.2 \cdot |1\rangle. \tag{14}$$

In this way, two mixed quantum numbers may be obtained, and in the same way, limits of variability of α and β can be determined from the dependence (12) and (14). In the discussed example, module α is determined from the interval below for the dominant state |0>:

$$0.80 \leq \alpha \leq 0.98, \tag{15}$$

and module β is determined based on superposition rule for a pure state |1>. Similarly, for the prevailing state |1> module β is determined from the interval:

$$0.20 \leq \beta \leq 0.60, \tag{16}$$

and module α is determined based on superposition rule for a pure state |0>.

2.4. QEA for the improvement of parameters of the neural model of PR–02 robot's arm movement

Quantum Evolutionary Algorithm was used, i.a., for the improvement of parameters of PR–02 robot's arm movement, taking basic assumptions of evolutionary algorithm [44] as an example, which was supplemented with elements of quantum computer science. Four parameters of PR–02 robot were distinguished in the model [28,33], i.e.:

θ_1 – angle of rotation of the whole platform of PR–02 robot (values from the interval –300°– 0°),

λ_2 – height of mounting of PR–02 robot's arm platform (values from the interval 0 – 0.200 m),

λ_3 – length of PR–02 robot's arm reach (values from the interval 0.376 – 0.676 m),

θ_4 – angle of rotation of the end of robot's arm (values from the interval 0° – 360°) – this element may make a full rotation around its own axis but does not have impact on the position of the end of the robot's arm.

By changing the four parameters D–H described above, one may directly control the movement of PR–02 robot's arm by lifting the main arm at the maximum height of 0.200 m, performing incomplete rotation around its own axis – up to 300°, pulling out the robot's arm at the length of 0.300 m, performing full rotation around the axis of arm's end.

QEA algorithm consists of the following basic steps:

1) Fetching four parameters [Θ_1, λ_2, λ_3, Θ_4] as elements of analytic model of the movement of PR–02 robot's arm's end, being input quantities (control) of the artificial neural network (ANN), i.e. parameters controlling a neural model of the PR–02 robot.
2) Generating initial population (IP) for the determined interval of accuracy of the parameters of the model.
3) Determination of values of individual chromosomes based on robustness (adaptation) function using Euclidean distance of output parameters (x, y, z) ANN.
4) Performing selection of individuals, e.g. by applying tournament selection method for further processing using adaptation function (best adapted individuals are those whose point is located closest to the mean distance of all points.
5) Using genetic operators (e.g. single–point crossover, mutation connected with changing the value of the gene by one order of magnitude).

Robustness (Adaptation) function is the Euclidean distance, and takes the following form:

$$\Delta = \sqrt{(X - \bar{x})^2 + \quad (Y - \bar{y})^2 + \quad (Z - \bar{z})^2} \tag{17}$$

where: Δ – distance between the location of a given point representing a certain chromosome and a point being a mean value of all points (chromosomes),

X – the value of component X, Y – the value of component Y, Z – the value of component Z, $\overline{x}, \overline{y}, \overline{z}$ – means of respective components.

As a result, a chromosome with better parameters may be generated in each successive epoch. Iterations are repeated until the assumed improvement regarding the location of the point of PR–02 robot's arm is obtained, followed by the correction of the location of the next point performed in the same way. Quantum computer science was used in computations.

3. Analysis of possibilities of impelentation of QEA

Analysis of possibilities of implementation of QEA was performed from the viewpoint of solutions supplied by MATLAB and Simulink environment using Matlab language, System Optimization Toolbox and m–files. Special attention was drawn to the structure of QEA, in which quantum information is processed. A number of notions, necessary for the appropriate implementation of QEA were defined, and a method of performing quantum calculations, and their realization using quantum gates and circuits was shown based on concrete numeric examples. The elaboration of the above fundamentals of quantum computer science involved, i.a., the possibilities of implementation of QEA for the improvement of parameters of the model of PR–02 robot's movement published in works [32–33] and attempts at the implementation of the model for the improvement of TGEE model published in works [35–36]. Possibilities of improvement of parameters of ANN, including the values of weights in both layers result from the increase of the degree of accuracy of the problem being solved, in accordance with the increased precision inspired by methods of quantum computer science and mechanics.

The problem of mixed numbers was described, i.a., in work [32], where a method of obtaining quantum numbers was proposed. By applying the above mentioned method of obtaining mixed quantum numbers and fundamentals of Quantum Evolutionary Algorithm, a method of performing quantum calculations using classic computers was developed, which involves the following basic steps:

1) changing a real number (values of weights of the ANN) written in a decimal system into a binary number,
2) assuming that a binary value 1 is a pure state |1> of a quantum number, and a binary value 0 is a pure state |0> of a quantum number,
3) determining a mixed state using the method described in point III for each single qubit, or for a qubit representing all pure states of a real number based on the binary number,
4) generating a structure of a model of chromosomes for the initial population (IP) that consists of a pair of chromosomes, one of which represents the lower limit and another one the upper limit of individual

mixed numbers (structure composed of two two–dimensional chromosomes with the number of qubits corresponding to the length of a binary number corresponding to the real number being the value of the weight),

5) generating IP in the number of chromosomes appropriate for the task being solved by random selection of values of genes representing weights from the intervals represented by the lower limit and the upper limit of each mixed number,
6) determination of a quantum value of each chromosome by using a model of the ANN system – dependence (1),
7) sorting and coupling chromosomes according to their values (phenotype),
8) crossing–over coupled chromosomes with probability p_k,
9) performing mutation within IP with probability p_m,
10) determination of robustness as a discrepancy between a quantum value of each chromosome from the population and mean value of all chromosomes and using it in the process of selection of chromosomes for further processing,
11) performing selection according to the adopted method of selection adjusted to quantum calculations,
12) repeating the computational process until the value of discrepancy (of robustness function) is smaller than the assumed degree of accuracy of the system model determined by the distance of the obtained solution from the mean solution.

Example 2. Determination of quantum numbers

Let us determine a quantum value of the adder net_1 with 24 – inputs signals as a subsystem of ANN (u_1 … u_{24}, here: volume of ee in individual hours of the 24 hour day), one output signal (y_1, here: average price of energy ee sold) and $w_{1,1}$ … $w_{1,24}$ – weights of layer **L1** connected with neuron 1.

Step 1. Conversion of values written in decimal system for input u_1 and weight $w_{1,1}$ (here: for hour 0–1), and for the first training pair (here: 1st January 2015) into binary numbers (a rule was adopted that the dimension of the state vector in Hilbert's space is equal to the dimension of weights values) [5,37], where:

decimal values are equal: ud_1=0.0345, $wd_{1,1}$=0.287526,

and **binary numbers** are equal: ub_1=[0.000010001101], $wb_{1,1}$=[0.010011100001], respectively,

Step 2. Conversion of values written in binary system for input u_1 and weight $w_{1,1}$ (here: for hour 0–1), and for the first training pair (here: 1st January 2015) into values in the quantum system a rule was adopted that the dimension of the state vector in Hilbert's space is equal to the dimension of weights values) respective:

binary values are aqual: ub_1=[0.000010001101] and $wb_{1,1}$=[0.010011100001],

and **quantum values** are aqual (assuming the weight of the added value, simplified recording to two decimal places and omitted part of the overall responsibility for rotation):

$$uk_1 = \begin{bmatrix} 0.87 & 1 & 0.77 & 0.98 & 0.67 & 0.74 & 0.89 & 0.85 & 0.4 & 0.37 & 0.77 & 0.44 \\ 0.49 & 0 & 0.64 & 0.20 & 0.74 & 0.67 & 0.46 & 0.53 & 0.99 & 0.93 & 0.64 & 0.90 \end{bmatrix},$$

$$wk_{1,1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.51 & 0.49 & 0.52 & 0.40 & 0.38 & 0.49 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.49 & 0.51 & 0.48 & 0.60 & 0.62 & 0.51 \end{bmatrix}. \tag{18}$$

Step 3. Determination of quantum value for the adder for neuron 1 of hidden layer in accordance with model (1) obtaining e.g. for input 1 (assuming the binary reference system positions quantum number resulting from individual qubits):

$$net_{1,1}^1(u) = \begin{bmatrix} 0{,}076504 & 0{,}001753 \\ 0{,}376192 & 0{,}124301 \end{bmatrix}, \tag{19}$$

and for all 24 inputs:

$$net_1^1(u) = \begin{bmatrix} 0.0646 & 0.7992 \\ 0.0902 & 0.7260 \end{bmatrix}. \tag{20}$$

Therefore, after elimination of the integer part of the value of elements responsible for the rotation, one may observe that respective individual elements of the matrix in the expression (20) are equal:

$$\alpha_1\alpha_2 = 0.0646,\ \alpha_1\beta_2 = 0.7992,\ \beta_1\alpha_2 = 0.0902,\ \beta_1\beta_2 = 0.7260. \tag{21}$$

Step 4. Determination of quantum value for the output of neuron 1 of the hidden layer by substituting the expression (14) to the dependence (**1**):

$$|y_1^1\rangle = f(|net_{i1}^1\rangle) = \frac{1}{1+e^{-2\begin{bmatrix}1\\0\end{bmatrix}\cdot\begin{bmatrix}0.0646 & 0.7992\\ 0.0902 & 0.7260\end{bmatrix}}} = \frac{1}{1+e^{-2\cdot|0\rangle\cdot|l_{m1}\rangle\ |l_{m2}^t\rangle}}, \tag{22}$$

where:

$|l_{m1}\rangle = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix}$ – the first mixed number (first quantum mixed number with the value resulting from the synthetized value of the weights vector,

$l_{m2}^t = \begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix}^t = [\alpha_2 \quad \beta_2]$ – the second mixed number (second quantum mixed number with the value resulting from the synthetized value of the input vector,

Therefore, based on the dependence (21) and superposition rule, values of both mixed numbers may be determined, whose respective values are equal:

$$\begin{aligned} l_{m1} &= \alpha_1|0\rangle + \beta_1|1\rangle = 0.7561|0\rangle + 0.6544|1\rangle, \\ l_{m2} &= \alpha_2|0\rangle + \beta_2|1\rangle = 0.0805|0\rangle + 0.9968|1\rangle, \end{aligned} \tag{23}$$

And the expression (22) may be transformed to the following form:

$$|y_1^1\rangle = f(|net_{i1}^1\rangle) = \frac{1}{1+e^{-2\left[|0\rangle\cdot\left[\begin{bmatrix}\varphi_{\alpha 1}\\ \varphi_{\beta 1}\end{bmatrix}+\begin{bmatrix}0.7561\\ 0.6544\end{bmatrix}\ \begin{bmatrix}\varphi_{\alpha 2}\\ \varphi_{\beta 2}\end{bmatrix}+\begin{bmatrix}0.0805\\ 0.9968\end{bmatrix}\right]}}, \tag{24}$$

where: φ_1 and φ_2 are rotation vectors of both modules in each mixed number.

The value of expression (24) is the input value to layer 2 of the ANN, which may be written in the complex form, or in Dirac's notation.

Example 3. Crossover

Two subsequent quantum parent chromosomes, e.g. for the weight between net_l and input u_1:

$$w_{k1,1}^{1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.511 & 0.499 & 0.521 & 0.401 & 0.389 & 0.491 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.499 & 0.518 & 0.487 & 0.602 & 0.625 & 0.514 \end{bmatrix}.$$

$$w_{k1,1}^{2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.519 & 0.498 & 0.525 & 0.406 & 0.388 & 0.499 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.493 & 0.511 & 0.485 & 0.604 & 0.621 & 0.519 \end{bmatrix}. \quad (25)$$

Let us assume simple single–point crossover. An intersection point is selected at random, and parts of the two chromosomes are exchanged in order to obtain two descendant chromosomes (e.g. using Fredkin's gate, as a result of which two successive bits at the output are changed if the first input bit is equal 1, which may be applicable to single cubits, such as:

$$w_{k1,1}^{1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.511 & 0.498 & 0.525 & 0.406 & 0.388 & 0.499 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.499 & 0.511 & 0.485 & 0.604 & 0.621 & 0.519 \end{bmatrix}.$$

$$w_{k1,1}^{2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.519 & 0.499 & 0.521 & 0.401 & 0.389 & 0.491 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.493 & 0.518 & 0.487 & 0.602 & 0.625 & 0.514 \end{bmatrix}. \quad (26)$$

Example 4. Mutation

Operator of quantum mutation changes quantum state |1> into |0> and vice versa, and in the case of a mixed number it changes the amplitude connected with |1> into the amplitude connected with |0> and vice versa (using quantum gate NOT[4]), e.g. as a result of mutation in the case of the first matrix of weights, the following is obtained:

$$w_{k1,1}^{1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0.511 & 0.498 & 0.525 & 0.406 & 0.388 & 0.499 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0.499 & 0.511 & 0.485 & 0.604 & 0.621 & 0.519 \end{bmatrix}. \quad (27)$$

Example 5. Selection

Tournament selection method was adopted. A random selection of n quantum chromosomes (individuals) without replacement from a Population of Chromosomes (PCh), followed by selection of the best chromosome and copying it to the parent population (PP), and after the operation of copying, all selected chromosomes are returned to the PCh. Tournament is organized S times in order to obtained a complete TP of the desired size. Assessment of values of chromosomes is performed using robustness function determined as a discrepancy between value $y_1(ch_i)$ and the mean vale of the selected

[4] So called Pauli's gate X , for which unitary matrix is equal:

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

chromosomes (using Toffoli's quantum gate and Fanout's quantum gate for quantum calculations).

Example 6. Stop condition for QEA

Quantum evolutionary algorithm stops when it reaches quantum discrepancy below the pre–specified state of quantum discrepancy.

4. CONCLUSIONS

The Quantum Evolutionary Algorithm proposed in this work, which was verified in practice using PR–02 robot movement as an example, implemented in Matlab language, and verified in MATLAB and Simulink environment.

Improvement in parameters of PR–02 robot's arm's end movement was obtained, thereby indicating the direction of improvement of accuracy of the trajectory of the robot's movement.

A model of Quantum Evolutionary Algorithm of PR–02 robot's movement, its implementation and the results obtained are presented in chapter "Studying and implementation of QEA of the movement of PR–02 robot's arm" [32–33].

References

[1] Albert F.Y.C., Koh S.P., Chen C.P., Tiong S.K., Edwin S.Y.S., Optimizing Joint Angles of Robotic Manipulator Using Genetic Algorithm. IACSIT Press, wyd. 2, Singapur 2011.

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

[3] Bellac L. M., Admission to quantum computer science (in polish: Wstęp do informatyki kwantowej), WN PWN, Warszawa 2011.

[4] Embedded MATLAB™ User's Guide © COPYRIGHT 2007 by The MathWorks, Inc. Natick, MA 01760–2098 (USA).

[5] Duda R., The concept of dimension (in polish: O pojęciu wymiaru), Biblioteka matematyczna, PZWSz, Warszawa 1972.

[6] Giaro K., Kamiński M., Introduction to quantum computer science (in polish: Wprowadzenie do algorytmów kwantowych), AOW EXIT, Warszawa 2003.

[7] Giergiel J., Hendzel Z.: Robust algorithms for motion control follower wheeled mobile robots (in polish: Krzepkie algorytmy sterowania ruchem nadążnym mobilnych robotów kołowych), Materiały: Warsztaty mobilnych robotów, IPPT PAN, Warszawa, 1997, 65–72.

[8] Greenberg Y.S., Kochmanski M., Basics of quantum computing. Elementary introduction. (in polish: Podstawy informatyki kwantowej. Elementarne wprowadzenie).Wyd. URz., Rzeszów 2011.

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

[10] Helt P., Parol M., Piotrowski P., Methods of artificial intelligence in the electric power engineering (in polish: Metody sztucznej inteligencji w elektroenergetyce). OW PW. Warszawa 2000.
[11] Heller M., Elements of quantum mechanics for philosophers (in polish: Elementy mechaniki kwantowej dla filozofów), OBI, Wyd. DT Biblios, Kraków 2011.
[12] Hervensalo M., Quantum algorithms (in polish: Algorytmy kwantowe), WSiP, Warszawa 2004.
[13] Hendzel Z., Żylski W.: The burly observer generalized speed mobile robot pie (in polish: Krzepki obserwator uogólnionych prędkości mobilnego robota kołowego), Materiały VI Krajowej Konferencji Robotyki, Wrocław 1998.
[14] Ho S.L., Yang S., Ni P., Huang J., A Quantum–Inspired Evolutionary Algorithm for Multi–Objective Design, IEEE Transactions on Magnetics, 2013, Volume: 49, Issue 5, pp.1609 – 1612.
[15] Korbicz J., Artificial neural networks and their application in electrical and power engineering (in polish: Sztuczne sieci neuronowe i ich zastosowanie w elektrotechnice i energetyce). Przegląd Elektrotechniczny nr 9/2009, s. 194–200.
[16] Kaczorek T., Dzieliński A., Dąbrowski W., Łopatka R., Fundamentals of control theory (in polish: Podstawy teorii sterowania). WNT, Warszawa 2007.
[17] Kwaśnicka H., Evolutionary computing in artificial intelligence (in polish: Obliczenia ewolucyjne w sztucznej inteligencji), PWr., Wrocław 1999.
[18] Kasabov N., Neuro–, Genetic–, and Quantum Inspired Evolving Intelligent Systems, 2006 International Symposium on Evolving Fuzzy Systems, 2006, pp. 63 – 73.
[19] Kłopotek M. A., Technical application of evolutionary strategies (in polish: Techniczne zastosowanie strategii ewolucyjnych). IPI PAN, Warszawa 1999.
[20] Li J., Li J., Next–Day Electricity Price Forecasting Based on Support Vector Machines and Data Mining Technology. Proceedings of the 27th Chinese Control Conference, Kunming, Yunnan, China 2008, pp. 630 – 633.
[21] Liao G., Using chaotic quantum genetic algorithm solving environmental economic dispatch of Smart Microgrid containing distributed generation system problems, Power System Technology, POWERCON, 2010 International Conference on, 2010, pp.: 1 – 7.
[22] Liu Yuanyuan Y., Li M., A bidding method of power market based on immune genetic algorithm, 2008 27th Chinese Control Conference, 2008, pp. 28 – 32.
[23] Michalewicz Z., Genetic Algorithms + Data Structures = Programs evolutionary (in polish: Algorytmy genetyczne + struktury danych = programy ewolucyjne). WNT, Warszawa 1999.
[24] Narayanan A., Moore M., Quantum–inspired genetic algorithms, Proceedings of IEEE International Conference on Evolutionary Computation, pp. 61–66, IEEE, 1996.
[25] Osowski S., Neural networks for information processing (in polish: Sieci neuronowe do przetwarzania informacji), PW, Warszawa 2000.
[26] Pittenger A. O., An Introduction to Quantum Computing Algorithms, Birkhauser, Boston 2000.

[27] Sawerwain M., Wisniewska J., Quantum computing. Selected circuits and algorithms (in polish: Informatyka kwantowa. Wybrane obwody i algorytmy), WN PWN SA, Warszawa 2015.
[28] Szkodny T.: Kinematics of industrial robots (in polish: Kinematyka robotów przemysłowych). Wyd. PŚ., Gliwice 2013.
[29] Susskind L., Friedeman A., Quantum mechanics. theoretical minimum (polish: Mechanika kwantowa. Teoretyczne minimum), Prószyński i S–ka, Warszawa 2016.
[30] Tadeusiewicz R., Sztuczne sieci neuronowe, AOW RW, Warszawa 1993,
[31] Tadeusiewicz R., Szaleniec M., Lexicon neural networks (in polish: Leksykon sieci neuronowych). Wyd. Fundacji „Projekt Nauka", Wrocław 2015.
[32] Tchórzewski J., The possibility of quantum computing to improve the accuracy of modeling. Part 1. Quantum evolutionary algorithm (in polish: Możliwości informatyki kwantowej do poprawy dokładności modelowania. Część 1. Kwantowy algorytm ewolucyjny). Poznan University of Technology. Academic Journal. Wyd. PP, Poznań 2016.
[33] Tchórzewski J., Wołynka Ł., The possibility of quantum computing to improve the accuracy of modeling. Part 2. Quantum evolutionary algorithm motion robot PR–02 (in polish: 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 2016.
[34] Tchórzewski J., Chyży E., Researching the Development of the Electrical Power System Using Systemically Evolutionary Algorithm. International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 6, No. 1, pp. 1–10, 2016.
[35] Tchórzewski J., Ruciński D., Modeling neural safe development TGE using DAM quotations (in polish: Modelowanie neuronalne bezpiecznego rozwoju TGEE z wykorzystaniem notowań RDN), Rozdział w książce pt. Wymiary zarzadzania ryzykiem w obrocie gospodarczym. Wydawnictwo CeDeWu, Warszawa 2016.
[36] Tchórzewski J., Ruciński D., Quantum Inspired Evolutionary Algorithm to Improve Parameters of Neural Models on Example of Polish Electricity Power Exchange. EPNet 2016 Electric Power Network, Xplore Digital Library.
[37] Tchórzewski J., The development of the power system in terms of control theory and systems (in polish: Rozwój systemu elektroenergetycznego w ujęciu teorii sterowania i systemów). OW PWr. Wrocław 2013.
[38] Tchórzewski J., Systemic Evolution Algorithms for the design of the development of the electricity market, 10th International Conference on the European Energy Market, EEM 2013, IEEE Xplore Digital Library, 2013, pp. 1–8.
[39] Tchórzewski J., Jurkowski B., The condition and the possibility of implementing quantum information on the example of a quantum evolutionary algorithm (in polish: Stan i możliwości implementacyjne informatyki kwantowej na przykładzie kwantowego algorytmu ewolucyjnego), [w:] Postępy w Elektrotechnice Stosowanej. Materiały VII Ogólnopolskiej Konferencji Naukowo–Technicznej PES–7. OW PTETiS. Wydz. Elektryczny PW, Ośrodek Promocji Badań z/z Energoelektroniki PW. PS Sekcja IEEE. Warszawa–Kościelisko 2009.

[40] Tchórzewski J., Wąsowski A., System SAE evolutionary algorithm in generating the electricity market (in polish: Systemowy algorytm ewolucyjny SAE w generowaniu rozwoju rynku energii elektrycznej). Materiały V Międzynarodowej Konferencji Naukowo–Technicznej ENERGETYKA. PWr., Wrocław 2008.
[41] Tchórzewski J., Systemic Method of Structure and Parameters Researching for Beginning Population Building of Evolving Algorithm SAE on the Example of Electricity Market, Polish Journal of Environmental Studies, Vol. 17, No 2A, 2008, pp. 86–89.
[42] Tchórzewski J., Owczarczyk P., Model and Implementation of Quantum Neural Network for Transfer Knowledge about Electricity Market, Studia Informatica, Vol 3–4 (7) 2007, AP, Siedlce 2007.
[43] Tchórzewski J., Owczarczyk P., Model and implementation of quantum neural networks for processing the knowledge of the electricity market (in polish: Model i implementacja kwantowej sieci neuronowej do przetwarzania wiedzy o rynku energii elektrycznej). Computer Application in Electrical Engineering, Politechnika Poznańska, Poznań 2007.
[44] Tchórzewski J., Systemowy Algorytm Ewolucyjny SAE. Bio–Algorithms and Med–Systems. Journal Edited by Collegium Medicum. Vol. 1. No. 1/2/2005. UJ. Kraków 2005, pp. 61–64.
[45] Tchórzewski R. J, Ruciński A., Model and implementation of a system of evolutionary algorithm to search for a new system state electricity grid (in polish: Model i implementacja systemowego algorytmu ewolucyjnego do poszukiwania nowego stanu systemu elektroenergetycznej sieci przesyłowej). Studia Informatica, Systemy i technologie informacyjne. UPH, Siedlce 2005, nr 1 (5), s. 59–67.
[46] Tchórzewski J., System testing the correctness of the development of the control system on the example of the development of power transmission network (in polish: Systemowe badanie prawidłowości rozwoju systemu sterownia na przykładzie rozwoju elektroenergetycznej sieci przesyłowej). Rozprawa naukowa nr. 58. AP. Siedlce 1997.
[47] Tchórzewski J., Cybernetics and systems development life (in polish: Cybernetyka życia i rozwoju systemów). Monografie nr. 22. Wyd. WSRP w Siedlcach. Siedlce 1992.
[48] Tchórzewski J., Engineering systems development (in polish: Inżynieria rozwoju systemów). Monografie nr 18. Wyd. WSRP w Siedlcach. Siedlce 1990.
[49] Węgrzyn S, Klamka J, i inni, Nano and quantum computing systems (in polish: Nano i kwantowe systemy informatyki), Wyd. PŚ, Gliwice 2004.
[50] Wierzchoń S., Kłopotek M., Cluster analysis. Monografie. Wyd. IPI PAN. Warszawa 2015.
[51] Zieliński J. [red.], Intelligent systems in management. Theory and practice (in polish: Inteligentne systemy w zarządzaniu. Teoria i praktyka). WN PWN. Warszawa 2000.
[52] Żurada J., Barski M., Jędruch W., Artificial neural networks. Fundamentals of theory and application (in polish: Sztuczne sieci neuronowe. Podstawy teorii i zastosowania), PWN, Warszawa 1996.

(Received: 28. 09. 2016, revised: 17. 11. 2016)