doi

UDC 004.023, 539.18

V.M. Tkachuk, Ph.D. (Phys.-Math.), **M.I. Kozlenko,** Ph.D (Eng.),
M.V. Kuz Dr.Sc. (Eng.), **I.M. Lazarovych** Ph.D. (Eng.), **M.C. Dutchak**
Vasyl Stefanyk Precarpathian National University
(57 Shevchenko str., Ivano-Frankivsk, 76018, Ukraine,
e-mail: tkachukv@gmail.com)

Function Optimization Based on Higher-Order Quantum Genetic Algorithm

Quantum genetic algorithms (QGA) are typically built using the traditional representation of the quantum chromosome in the form of system of independent qubits. This makes it impossible to use a very powerful quantum calculations mechanism, namely quantum state entanglement. In this paper we implement a higher-order QGA and illustrate efficiency of the algorithm on the basis of example of optimization problem solved for a test functions set. An adaptive quantum gate operator, which does not require a lookup table is also proposed. In comparison to traditional QGA, the transition to higher (more than two) orders in the algorithm implementation shows much better results in terms of the running time, convergence speed and solution precision.

K e y w o r d s: function optimization, quantum state entanglement, quantum genetic algorithm, quantum computation, quantum register.

Introduction. Quantum genetic algorithm is a relatively new evolutionary algorithm, which combines quantum computation ideas and the technology of classical genetic algorithms (cGA) [1, 2]. The probabilistic mechanism of the quantum computations, combined with the traditional evolutionary algorithm, provides the global search ability with good convergence speed and small population size. If the operators of the classical genetic algorithm are commonly known and have a clear physical interpretation by analogy with the biological mechanisms of the evolution, the quantum operators require a more detailed examination, physical interpretation and a mathematical formalization during the implementation [3, 4].

QGA does not require a quantum computer for implementation and is effective for an extensive range of scientific and engineering problems, which require approximate solutions with the minimized search time [5—9].

To enhance the effectiveness of the algorithm, we can supplement it with the genetic operators, inherent in the traditional GA [2, 10—12]. Taking into consideration the small size of the population, a quantum disaster operation can also be used in QGA to widen the search area and escape the local minimums [13].

The main QGA concepts were proposed by Narayanan and Moore [14]. During the quantum computations implementation the basic unit of information is a qubit — a quantum system, which may be in the $|0\rangle$ basis state or the $|1\rangle$ basis state. Quantum nature of the qubit lies in the superposition principle, under which the qubit generally is in a state, which is a linear combination of basis states:

$$|q\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

with a normalization constraint:

$$\alpha_0^2 + \alpha_1^2 = 1.$$

The quantum part of the information is in α_0^2 and α_1^2: values α_0^2, α_1^2 are the probability amplitudes of the qubit being in the states $|0\rangle$ and $|1\rangle$, respectively. The capacity of the quantum computations is caused by two accounts: quantum parallelization, which is based on the superposition principle, and states entanglement. The conventional QGA implementation utilizes the superposition principle only. The quantum chromosome is formed as a structured set of independent qubits. For example, if it consists of $N = 16$ qubits, it can be schematically illustrated as follows:

$\|q_1\rangle$	$\|q_2\rangle$	$\|q_3\rangle$	$\|q_4\rangle$	$\|q_5\rangle$	$\|q_6\rangle$	$\|q_7\rangle$	$\|q_8\rangle$	$\|q_9\rangle$	$\|q_{10}\rangle$	$\|q_{11}\rangle$	$\|q_{12}\rangle$	$\|q_{13}\rangle$	$\|q_{14}\rangle$	$\|q_{15}\rangle$	$\|q_{16}\rangle$

Higher-order QGA. A higher-order QGA algorithm was proposed in [15], where its efficiency for combinatorial optimization problems was proven. Paper [16] illustrates its application for Topology Control of wireless sensor network problem. The overall ideology of the algorithm is similar to the regular QGA, and the difference lies in the quantum operators implementation, which take into account the quantum chromosome representation as a quantum register set with entangled states.

Structure of Quantum Chromosomes. If each two qubits are entangled, then the chromosome can be represented as follows:

R_1		R_2		R_3		R_4		R_5		R_6		R_7		R_8	
$\|q_1\rangle$	$\|q_2\rangle$	$\|q_3\rangle$	$\|q_4\rangle$	$\|q_5\rangle$	$\|q_6\rangle$	$\|q_7\rangle$	$\|q_8\rangle$	$\|q_9\rangle$	$\|q_{10}\rangle$	$\|q_{11}\rangle$	$\|q_{12}\rangle$	$\|q_{13}\rangle$	$\|q_{14}\rangle$	$\|q_{15}\rangle$	$\|q_{16}\rangle$

where each quantum register R_i consists of two qubits ($r = 2$), which are in a superposition state. The number of all possible states n of such register is:

$$n = 2^r = 2^2 = 4,$$

and the quantum register has four basis states:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle.$$

There classical equivalent to the majority of the possible quantum register values simply does not exist. Unlike the classic register, these basis states are not the limit of all the possible values of the quantum register because of the superposition principle, according to which the entangled system of two qubits can be in the state, which is a linear combination of the basis states:

$$|q\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle.$$

Here $\alpha_0^2, \alpha_1^2, \alpha_2^2$ and α_3^2 represent the probabilities of the quantum register being in the corresponding state. When using the quantum states entanglement, the size of the matrix M, which is required for the quantum chromosome representation for $r = 2$, remains the same. During the implementation on a regular computer, for a one qubits representation we need two elements of the matrix, so:

$$M = 2 \cdot N = \frac{N}{2} \cdot 2^r = \frac{N}{2} \cdot 4 = 32.$$

It is convenient to use the following structure for representing one individual during the QGA implementation. It consists of $k = N / r$ quantum registers, as demonstrated in the table below (Table 1).

Plurality $\{\alpha_0^i, \alpha_1^i, \alpha_2^i, \alpha_3^i\}$ determines the state of the quantum register R_i with $n = 4$ basis states, and the set of k registers forms one individual in the population. The initial state of the qubit does not contain information about the problem or any of the characteristics of its solution. Therefore, the state of the quantum register does not contain the information too. So, the easiest way to initialize it is to set all the amplitudes α_n^i ($n \in 0,1,2,3$) to be equal to one another [1]. This means that after the initialization each register will be in the following state:

$$|q\rangle = \frac{1}{\sqrt{4}}|00\rangle + \frac{1}{\sqrt{4}}|01\rangle + \frac{1}{\sqrt{4}}|10\rangle + \frac{1}{\sqrt{4}}|11\rangle.$$

Table 1

R_1	R_2	R_3	…	R_i	…	R_k
α_0^1	α_0^2	α_0^3	…	α_0^i	…	α_0^k
α_1^1	α_1^2	α_1^3	…	α_1^i	…	α_1^k
α_2^1	α_2^2	α_2^3	…	α_2^i	…	α_2^k
α_3^1	α_3^2	α_3^3	…	α_3^i	…	α_3^k

After the transition to the higher-order quantum registers, or with the number of entangled qubits $r > 2$, the M matrix will be:

$$M = \frac{N}{r} \cdot 2^r > 2 \cdot N.$$

This means that for the quantum chromosome representation we additionally need to enlarge the size of the matrix in this case, both for representation of one individual and the population in general. Let us consider the quantum chromosome to consist of four quantum registers (R_1, R_2, R_3, R_4) and to be represented as follows:

R_1				R_2				R_3				R_4			
$\|q_1\rangle$	$\|q_2\rangle$	$\|q_3\rangle$	$\|q_4\rangle$	$\|q_5\rangle$	$\|q_6\rangle$	$\|q_7\rangle$	$\|q_8\rangle$	$\|q_9\rangle$	$\|q_{10}\rangle$	$\|q_{11}\rangle$	$\|q_{12}\rangle$	$\|q_{13}\rangle$	$\|q_{14}\rangle$	$\|q_{15}\rangle$	$\|q_{16}\rangle$

The total number of basis states for such register is equal to $2^4 = 16$. Namely, we have the following set:

$$|0000\rangle, |0001\rangle, |0010\rangle, ..., |1111\rangle.$$

According to the superposition principle, state of the register can be represented as follows:

$$|q\rangle = \alpha_0 |0000\rangle + \alpha_1 |0001\rangle + \alpha_2 |0010\rangle + ... + \alpha_{16} |1111\rangle.$$

The size of the matrix M for the quantum chromosome representation in this case equals:

$$M = \frac{N}{2} \cdot 2^r = \frac{16}{4} \cdot 2^4 = 64 > 2 \cdot N = 32.$$

During the QGA implementation, the size of the quantum chromosome N is determined by solution precision ε, the search area $[x_{\min}, x_{\max}]$ and the number of quantum register r basis states:

$$N = \log_{2^r} \left(\frac{x_{\max} - x_{\min}}{\varepsilon} + 1 \right).$$

The relation between the size of the matrix M and the size of the quantum register for search area $[-1,1]$ and $\varepsilon = 10^{-6}$ is shown in Fig. 1.

Quantum Chromosome Measurement. The solution of the problem is determined by the ending state of the quantum chromosome and can be obtained using the quantum measurement. The outcome of such measurement is the classical binary representation of the quantum chromosome. The quantum measurement operator is implemented in accordance to [15, 17].

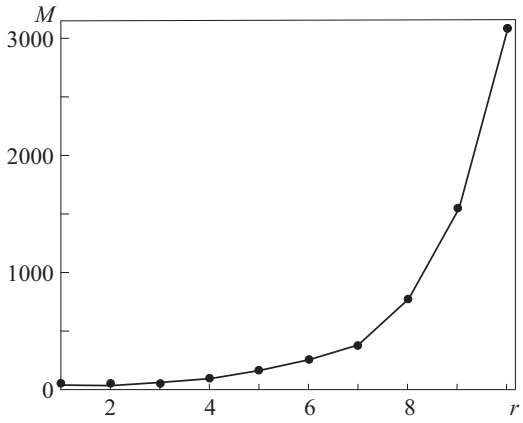

Table 2

State register index number	Classical representation of the register
0	000
1	001
2	010
3	011
4	100
5	101
6	110

Fig. 1. Relation between the size of the matrix M, which is used for representing one individual of the population, and the quantum register size r

The algorithm for measuring the state of the quantum chromosome, which consists of k quantum registers of size r, can be implemented in the following way:

A l g o r i t h m 1. Quantum register state measurement

```
1    for i ∈ 1,...,k do
2        rand ← random number in the area [0,1]
3            Sum ← 0
4            for j ∈ 0,...,2^r − 1 do
5                Sum ← Sum + [α^i_j]^2
6                if rand < Sum then
7                    p ← W_j
8                end if
9            end for
10       end for
```

Here the i index is responsible for going through the quantum registers R_i, and the j index determines the number of the state inside the quantum register itself. An auxiliary matrix W used during the algorithm implementation is listed in Table 2 for the case $r = 3$. It is used to convert the index number of the register state to the corresponding classical representation. In this case, classical representation of the register is just the state index number j in binary representation.

The result of applying the algorithm is its classical representation in accordance to the table of probabilities $(\alpha^i_0)^2, (\alpha^i_1)^2, (\alpha^i_2)^2, \ldots, (\alpha^i_{2^r-1})^2$ listed earlier.

Quantum Rotating Gates. For determining the solution, QGA changes the initial superposition of basis states by consecutively applying the quantum operators as the evolution progresses in time. All the information about the problem and its solution lay in the quantum gate, so its algorithm is the most important in the process of building any QGA. It takes the amplitudes of the quantum states to ensure that the normalization conditions are satisfied.

A change of one of the states influences all its other probabilities within the quantum register. This characteristic of the entangled states provides the parallelism of evaluations, which gives QGA an advantage over the regular quantum genetic algorithm.

The operator is applied to each of the registers R_i in two phases. In the first phase the probability amplitude of the selected quantum state b is enlarged:

$$(\alpha_i^b)' = \sqrt{[\alpha_i^b]^2 + \mu(1-\alpha_i^b)},$$

where μ — is a parameter in range $[0,1]$, which is determined based on the results of previous research. State b is determined by the representation of a fragment of the population's best chromosome, which corresponds to the quantum register R_i in the binary system. Expression also provides the fact that probability amplitude α_i^b can't be greater than 1. During the QGA implementation one fixed rotation angle is not enough for providing a good convergence speed of the algorithm, so adaptive behavior is also used for this purposes.

In the second phase, all the remaining probability amplitudes of the quantum register need to be decreased for preserving the normalization condition. To sum up, the algorithm of applying the operator to the quantum chromosome, which consists of k quantum registers of size r, can be represented as follows:

A l g o r i t h m 2. Quantum gate operator

1 **for** $i \in 1,\ldots,k$ **do**

2 $bestamp \leftarrow$ b

3 $Sum \leftarrow 1 - [\alpha_i^{bestamp}]^2$

4 $\alpha_i^{bestamp} = \sqrt{[\alpha_i^{bestamp}]^2 + \mu(1-\alpha_i^{bestamp})}$

5 $M \leftarrow \sqrt{\frac{1-[\alpha_i^{bestamp}]^2}{Sum}}$

6 **for** $amp \in \{0,1,2,\ldots,2^r-1\}$ **do**

7 **if** $amp \neq bestamp$ **then**

8 $\alpha_i^{amp} = M \cdot \alpha_i^{amp}$

9 **end if**

10 **end for**

11 **end for**

Therefore, in each next generation we increase the probability of generating the classical individuals most similar to the best one in the result of the measurement. It is also important that during the building of the quantum gate we do not require a lookup table, which is one the significant disadvantages of the QGA.

Simulation Test. QGA with adaptive quantum gate operator (aQGA) and higher-order quantum register system is implemented using C++ programming language, and the simulations are performed on an Intel Celeron CPU G1840 2.80GHz, 4.0 Gb RAM. A number of numerical optimization problems were analyzed to illustrate the algorithm design. The following test functions, taken from virtual library of Simulation Experiments in Simon Fraser University, were used:

Rosenbrocks valley—

$$F_1(x_1, x_2) = 100\,(x_1^2 - x_2^2) + (1 - x_1^2),\ -2{,}048 \le x_i \le 2{,}048,\ i = 1, 2.$$

Coldstein Price function —

$$F_2(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$$
$$\times [30 + (2x_1 + 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)],$$
$$-2{,}0 \le x_i \le 2{,}0,\ i = 1, 2.$$

Schaffer function —

$$F_3(x_1, x_2) = 0{,}5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0{,}5}{[1{,}0 + 0{,}001(x_1^2 + x_2^2)]^2},$$
$$-100{,}0 \le x_i \le 100{,}0,\ i = 1, 2.$$

Mono-pole and six-peak camelback function —

$$F_4(x) = 10 + \frac{\sin(1/x)}{0{,}1 + (x - 0{,}16)^2},$$
$$0{,}0 \le x_i \le 1{,}0.$$

Dual-pole and six-peak camelback function —

$$F_5(x_1, x_2) = \left(4 - 2{,}1x_1^2 + \frac{1}{3}x_1^4\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)\,x_2^2,$$
$$-3{,}0 \le x_i \le 3{,}0,\ i = 1, 2.$$

Multi-peak positive function —

$$F_6(x) = e^{-0{,}001x}\cos^2(0{,}8x),\ -2{,}0 \le x \le 5{,}0.$$

Trid function —

$$F_7(x_1, x_2, ..., x_d) = \sum_{i=1}^{d} (x_i - 1)^2 - \sum_{i=2}^{d} x_i x_{i-1}, \; -d \le x_i \le d, \; i = 1, 2, ..., d.$$

Levy function —

$$F_8(x_1, x_2, ..., x_d) = \sin^2(\pi\omega_1) + (\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)] +$$

$$+ \sum_{i=1}^{d} (\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)],$$

here $\omega_i = 1 + \frac{x_i - 1}{4}$, $-36 \le x_i \le 36$, $i = 1, 2, ..., d$.

Schwefel function —

$$F_9(x_1, x_2, ..., x_d) = 418{,}9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|}),$$

$$-500 \le x_i \le 500, \; i = 1, 2, ..., d.$$

Ackley function —

$$F_{10}(x_1, x_2, ..., x_d) = -20 \cdot e^{-0,2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{d} \cos(2\pi x_i)} + 22{,}71828,$$

$$-32{,}768 \le x_i \le 32{,}768, \; i = 1, 2, ..., d.$$

Rastrigin function —

$$F_{11}(x_1, x_2, ..., x_d) = 10 \cdot d - \sum_{i=1}^{d} [x_i^2 - 10 \cdot \cos(2\pi x_i)],$$

$$-5{,}12 \le x_i \le 5{,}12, \; i = 1, 2, ..., d.$$

Optimization and Results. Optimization of the above-mentioned eleven functions using GA, QGA and aQGA with different quantum register sizes is performed for the simulation test of the proposed algorithm. For the QGA implementation the following simulation parameters were used: the size of the population $s = 50$ in case if the number of parameters of the studied function $d = 6$; and $s = 10$ in case d does not exceed two; the number of evolution iterations over time $t = 500$; precision $\varepsilon = 1 \cdot 10^{-6}$, which are the same as those taken in aQGA. The parameters used for GA are identical to [18]: precision $\varepsilon = 1 \cdot 10^{-6}$, population size $s = 50$, crossover probability $pc = 0{,}8$, mutation probability $pm = 0{,}01$ and total generations of iteration $t = 500$.

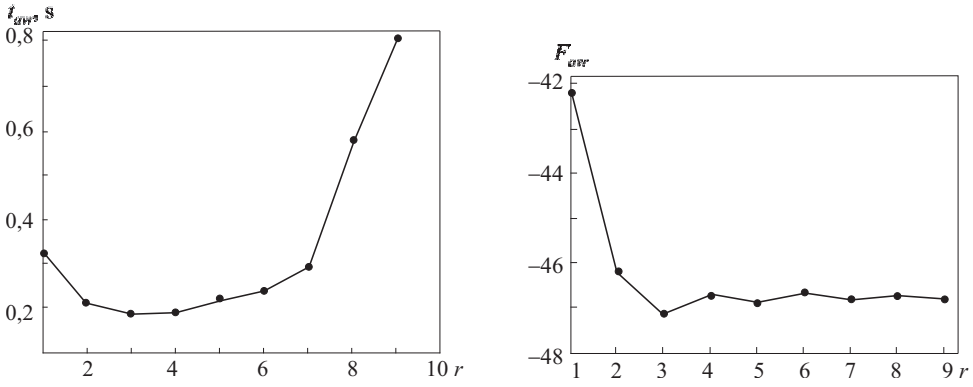

Fig. 2. Average algorithm running time t_{avr} as a function of quantum register size r for function F_7

Fig. 3. Average fitness of the best population individual F_{avr} as a function of quantum register size r for function F_7

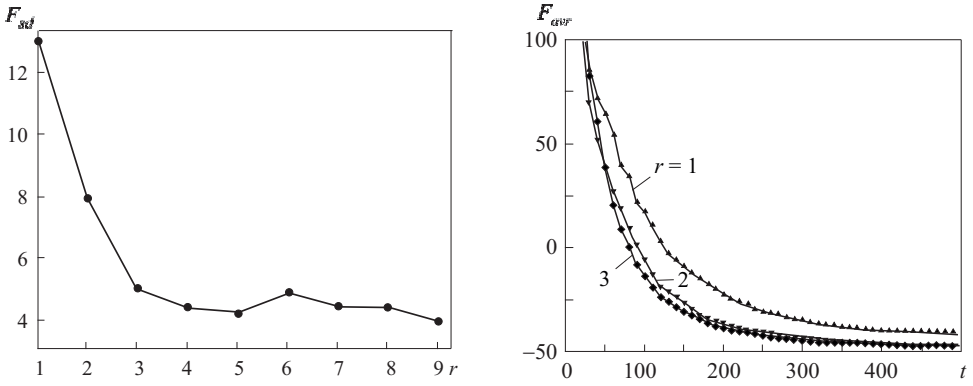

Fig. 4. Standard deviation F_{sd} of the average fitness in dependency of quantum register size r for function F_7

Fig. 5. Average for 100 runs population evolution of the population in time for F_7 for different values of the quantum register length r

To determine the optimal μ value, we have analyzed its influence on the average fitness of the best population individual. The parameters are evaluated as an averaging result after 1000 runs of aQGA, if not specified otherwise. It is important that the size of the quantum register has very little effect on the optimal μ value for this function. Taking into account the F_{avr} weak dependency on μ in the optimal values area, the values $\mu \approx 0{,}015$ for $d = 6$ and $\mu \approx 0{,}004$ for $d \le 2$ can be considered acceptable for all studied functions.

aQGA effectiveness evaluation. The effectiveness evaluation is performed based on two main parameters. The first is the running time for the standard set of the input parameters. For illustrative purposes, we will limit the quantum re-

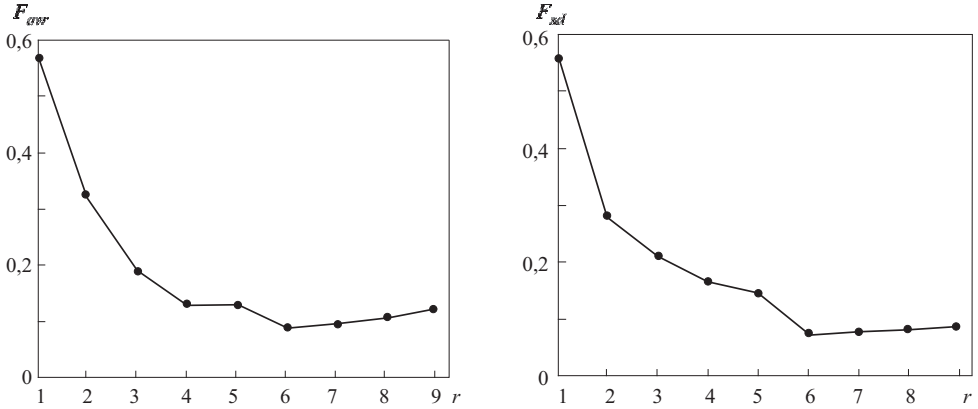

Fig. 6. Average fitness of the best population individual F_{avr} as a function of quantum register size r for function F_8

Fig. 7. Standard deviation F_{sd} of the average fitness as on Fig. 6 in dependency of quantum register size r for function F_8

gisters to $r = 6$ inclusive. There are no principal restrictions on the implementation of aQGA with $r > 6$. Taking into account the fact that the algorithm is implemented on a classical computer, the only principal constraint is the fast increase of the running time caused by the increase of the matrix size, which is required for the population individuals representation (see Fig. 1).

Fig. 2 illustrates the dependency between the average aQGA running time t_{avr} and the quantum register size r on the example of F_7 function optimization. It must be noted, that the algorithm running time almost does not depend on the test function, and is determined by the search area size with all the other simulation parameters values fixed (see Tab. 3).

Statistical results of optimization of test functions F_1— F_6, including the average value of the best population individual F_{avr}, its standard deviation, best result *Best* and worst result *Worst*, retrieved during the optimization of the lower-orders test functions, are listed in Tab. 4.

Table 3

F_i	cGA	QGA	aQGA					
			$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$
F_1	0,0362	0,0202	0,0168	0,0101	0,0097	0,0094	0,0106	0,0117
F_2	0,0497	0,0254	0,0208	0,0134	0,0126	0,0114	0,0104	0,0115
F_4	0,0175	0,0118	0,0079	0,0047	0,0042	0,004	0,0054	0,0058
F_6	0,0193	0,0121	0,0092	0,0058	0,0049	0,0044	0,0052	0,0056
F_8	0,872	0,51	0,322	0,215	0,179	0,181	0,206	0,241
F_{11}	0,773	0,491	0,273	0,168	0,143	0,14	0,16	0,199

Table 4

Variable	cGA	QGA	aQGA					
			$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$
				F_1				
F_{avr}	0,149	0,0155	0,00225	0,00185	0,00204	0,0035	0,0081	0,0105
F_{sd}	0,251	0,038	0,00322	0,00366	0,00421	0,0078	0,0146	0,0168
Best	0,052	0	0	0	0	0	0	0
Worst	2,139	0,583	0,0577	0,0513	0,0845	0,1201	0,133	0,0144
				F_2				
F_{avr}	3,752	3,013	3,0136	3,0029	3,0012	3,0004	3,0005	3,0003
F_{sd}	0,824	0,0451	0,0333	0,0056	0,0039	0,0007	0,0016	0,0007
Best	3,001	3	3	3	3	3	3	3
Worst	5,24	3,972	3,485	3,059	3,066	3,008	3,015	3,006
				F_3				
F_{avr}	0,134	0,0081	0,019	0,0102	0,0093	0,0088	0,0089	0,0111
F_{sd}	0,087	0,0093	0,0066	0,0056	0,0057	0,0035	0,0044	0,0086
Best	0	0	0	0	0	0	0	0
Worst	0,562	0,0329	0,0516	0,0379	0,0373	0,0372	0,0286	0,0299
				F_4				
F_{avr}	19,792	19,82	19,8949	19,8949	19,8949	19,8949	19,8949	19,8949
F_{sd}	$6{,}9\cdot10^{-2}$	$1{,}3\cdot10^{-2}$	$2{,}5\cdot10^{-5}$	0	0	0	0	0
Best	19,8944	19,8949	19,8949	19,8949	19,8949	19,8949	19,8949	19,8949
Worst	19,476	19,537	19,8941	19,8949	19,8949	19,8949	19,8949	19,8949
				F_5				
F_{avr}	–1,021	–1,03	–1,0303	–1,315	–1,315	–1,315	–1,316	–1,316
F_{sd}	$5{,}1\cdot10^{-3}$	$3{,}5\cdot10^{-3}$	$3{,}6\cdot10^{-3}$	$6{,}0\cdot10^{-4}$	$8{,}0\cdot10^{-4}$	$7{,}0\cdot10^{-4}$	0	0
Best	–1,0315	–1,0316	–1,0316	–1,0316	–1,0316	–1,0316	–1,0316	–1,0316
Worst	–1,097	–1,018	–1,0081	–1,0196	–1,0194	–1,0196	–1,0316	–1,0316
				F_6				
F_{avr}	–0,997	–1	–1	–1	–1	–1	–1	–1
F_{sd}	$2{,}2\cdot10^{-3}$	0	0	0	0	0	0	0
Best	–1	–1	–1	–1	–1	–1	–1	–1
Worst	–0,994	–1	–1	–1	–1	–1	–1	–1

Quantum state entanglement provides a higher level of variable correlation, and as a result, the efficiency of the algorithm increases [19]. This is illustrated by the results of the multipa- rameter test functions F_7 — F_{11} optimization.

The influence of the different size of the quantum register (quantum state entanglement) on the aQGA efficiency can be evaluated using the average fitness of the population best individual F_{avr} and the standard deviation of the aver-

Table 5

Variable	cGA	QGA	aQGA					
			$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$
				F_7				
F_{avr}	–22,7	–35,3	–42,17	–45,24	–47,38	–46,88	–47,34	–46,6
F_{sd}	50,3	20,7	13,37	10,1	5,004	4,2	3,61	5,07
Best	–45,4	–50	–50	–49,997	–50	–49,99	–50	–49,99
Worst	31,4	20,3	5,01	4,07	–3,38	–30,7	–30,1	–12,33
				F_8				
F_{avr}	2,53	1,02	0,585	0,329	0,207	0,125	0,111	0,056
F_{sd}	1,76	0,93	0,534	0,274	0,215	0,14	0,136	0,059
Best	0,52	0,15	0,016	$1{,}6\cdot10^{-3}$	$3{,}3\cdot10^{-4}$	$5{,}0\cdot10^{-4}$	$1{,}4\cdot10^{-4}$	$5{,}1\cdot10^{-4}$
Worst	6,44	2,32	2,82	0,981	0,961	0,721	0,77	0,402
				F_9				
F_{avr}	50,7	46,03	39,1	12,82	0,78	9,08	2,79	0,61
F_{sd}	86,7	74,2	53,6	21,45	3,91	17,45	8,28	1
Best	3,91	0,984	0,072	$8{,}0\cdot10^{-3}$	$2{,}6\cdot10^{-3}$	$1{,}3\cdot10^{-3}$	$2{,}6\cdot10^{-4}$	$9{,}0\cdot10^{-4}$
Worst	401,6	388,5	374,1	230,07	5,56	106,7	62,9	29,6
				F_{10}				
F_{avr}	1,05	0,138	0,174	0,064	0,112	0,227	0,288	0,559
F_{sd}	0,79	0,624	0,484	0,107	0,172	0,373	0,208	0,691
Best	0,92	$1{,}0\cdot10^{-3}$	$1{,}6\cdot10^{-3}$	$1{,}5\cdot10^{-3}$	$7{,}5\cdot10^{-4}$	$2{,}1\cdot10^{-3}$	$5{,}0\cdot10^{-3}$	$4{,}9\cdot10^{-3}$
Worst	6,42	1,97	3,3	1,61	1,77	2,29	1,38	3,23
				F_{11}				
F_{avr}	7,55	2,831	3,51	2,57	1,2	2,79	0,31	0,33
F_{sd}	3,12	2,725	2,16	1,23	1,69	1,15	0,42	0,49
Best	0,561	$1{,}8\cdot10^{-3}$	$1{,}9\cdot10^{-3}$	$1{,}7\cdot10^{-3}$	$5{,}9\cdot10^{-5}$	$6{,}2\cdot10^{-3}$	$2{,}0\cdot10^{-4}$	$6{,}2\cdot10^{-5}$
Worst	31,67	9,23	12,16	5,65	7,04	5,99	2,53	3,11

age fitness F_{sd} over 1000 algorithm runs. Fig. 3 and Fig. 4 list these values on the example of F_7 function optimization.

The advantages of the transition to higher orders in the QGA implementation are confirmed by the results in Fig. 5. The examples given in this paper prove that the bigger is the size of the quantum register with entangled states, the faster is the relaxation of the population to the optimal value, especially on the early stages.

We can observe a similar pattern when optimizing the function F_8, on Fig. 6 and Fig. 7.

Statistical results retrieved during the optimization of higher-order test functions F_7 — F_{11} of large dimension ($d = 6$) are listed in Tab. 5.

Conclusions

We proposed and implemented a new approach to solving numerical optimization problems. The approach uses the genetic algorithms technology and the quantum computations ideology, based on the higher-order quantum registers. The results of the simulation using a set of test functions illustrate better performance and efficiency of the suggested approach.

The performed simulations allow us to determine the optimal size of the quantum register to be equal to 3—5 qubits by analyzing the ratio of efficiency/performance. Because of the quantum registers state entanglement the algorithm running time has been almost halved, and in the same time the results were improved and the convergence of the algorithm was faster.

The quantum gate operator adopts the adaptive behavior of the quantum state rotation angle and does not require a lookup table.

REFERENCES

1. Han, K.-H. and Kim, J.-H. (2000), "Genetic quantum algorithm and its application to combinatorial optimization problem", *Proceedings of the 2000 Congress on Evolutionary Computation*, USA, 2, pp. 1354-1360.
2. Roy, U., Roy, S. and Nayek, S. (2014), "Optimization with quantum genetic algorithm", *International Journal of Computer Applications*, Vol. 102, no. 16, pp. 1-7.
3. Zhang, G. (2011), "Quantum-inspired evolutionary algorithms: a survey and empirical study", *Journal of Heuristics*, Vol. 2011, no. 17, pp. 303-351, DOI: 10.1007/s10732–010–9136–0.
4. Wang, H., Liu, J., Zhi, J. and Fu, C. (2013), "The Improvement of Quantum Genetic Algorithm and Its Application on Function Optimization", *Mathematical Problems in Engineering*, Vol. 2013.
5. Wang, L., Kowk, S.K. and Ip, W.H. (2012), "Design of an improved quantum-inspired evolutionary algorithm for a transportation problem in logistics systems", *Journal of Intelligent Manufacturing*, pp. 2227-2236, DOI: 10.1007/s10845–011–0568–7.

6. Jantos, P., Grzechca, D. and Rutkowski, J. (2012), "Evolutionary algorithms for global parametric fault diagnosis in analogue integrated circuits", *Bull. Pol. Tech*, Vol. 60, pp. 133-142, DOI:10.2478/v10175–012–0019–4.
7. Talbi, H., Batouche, M. and Draa, A. (2007), "A Quantum-Inspired Evolutionary Algorithm for Multiobjective Image Segmentation", *International Journal of Nuclear and Quantum Engineering*, Vol. 1, pp. 109-114.
8. Qin, C., Liu, Y. and Zheng, J. (2008), "A real-coded quantum-inspired evolutionary algorithm for global numerical optimization", *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1160-1164, DOI: 10.1109/ICCIS.2008.4670779.
9. Lin, D. and Waller, S. (2009), "A quantum-inspired genetic algorithm for dynamic continuous network design problem", Transportation Letters: *The International Journal of Transportation Research*, Vol. 1, pp. 81-93, DOI: 10.3328/TL.2009.01.01.81—93.
10. Malossini, A., Blanzieri, E. and Calarco, T. (2008), "Quantum genetic optimization", *IEEE Trans. Evol. Comput*, Vol. 12, pp. 231-241.
11. SaiToh, A., Rahimi, R. and Nakahara, M. (2014), "A quantum genetic algorithm with quantum crossover and mutation operations", *Quantum Information Process*, Vol. 13, pp. 737-755.
12. Lahoz-Beltra, R. (2016), "Quantum Genetic Algorithms for Computer Scientists", *Computers*, Vol. 5, no. 4, DOI: 10.3390/computers5040024.
13. Tkachuk, V. (2018). "Quantum Genetic Algorithm on Multilevel Quantum Systems", *Mathematical Problems in Engineering*, Vol. 2018, DOI: 10.1155/2018/9127510.
14. Narayanan, A. and Moore, M. (1996), "Quantum-inspired genetic algorithms", *Proceedings of the IEEE International Conference on Evolutionary Computation* (ICEC'96), Nagoya, Japan, pp. 61-66, DOI:10.1109/ICEC.1996.542334.
15. Nowotniak, R. and Kucharski, J. (2014), "Higher-Order Quantum-Inspired Genetic Algorithms", *Federated Conference on Annals of Computer Science and Information Systems*, Vol. 2, pp. 465-470, DOI: 10.15439/2014F99.
16. Ullah, S. and Wahid, M. (2015), "Topology Control of Wireless Sensor Network Using Quantum Inspired Genetic Algorithm", *International Journal of Swarm Intelligence and Evolutionary Computation*, Vol. 4, DOI: 10.4172/2090–4908.1000121.
17. Tkachuk, V. (2018), "Quantum Genetic Algorithm Based on Qutrits and Its Application", *Mathematical Problems in Engineering*, Vol. 2018, DOI :10.1155/2018/8614073.
18. Sun, Y. and Xiong, H. (2014), "Function Optimization Based on Quantum Genetic Algorithm", *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 7, no. 1, pp. 144-149, ISSN: 2040-7459; e-ISSN: 2040-7467.
19. Kuo, S.-Y. and Chou, Y.-H. (2017), "Entanglement-Enhanced Quantum-Inspired Tabu Search Algorithm for Function Optimization", *IEEE Access*, Vol. 5, pp. 13236-13252, DOI: 10.1109/ACCESS.2017.2723538.

Received 01.04.19

В.М. Ткачук, М.І. Козленко, М.В. Кузь, І.М. Лазарович, М.С. Дутчак

КВАНТОВИЙ ГЕНЕТИЧНИЙ АЛГОРИТМ ВИЩИХ ПОРЯДКІВ
В ЗАДАЧАХ ФУНКЦІОНАЛЬНОЇ ОПТИМІЗАЦІЇ

При побудові квантових генетичних алгоритмів (QGA) традиційним є представлення квантової хромосоми у вигляді системи незалежних кубітів. Це не дозволяє використати такий потужний механізм квантових обчислень, як заплутаність квантових станів. У роботі реалізовано QGA вищих порядків та проілюстровано його ефективність на прикладі задачі числової оптимізації з використанням ряду тестових функцій. Також запро-

поновано оператор квантового гейту із адаптивним характером роботи, що не вимагає використання таблиці пошуку. У порівнянні із традиційним QGA перехід до вищих (більше двох) порядків при реалізації алгоритму показує значно кращі результати як по часу виконання, так і по швидкості збіжності та точності знайденого розв'язку.

К л ю ч о в і с л о в а: функціональна оптимізація, заплутаність квантових станів, квантовий генетичний алгоритм, квантові обчислення, квантовий регістр.

В.М. Ткачук, Н.И. Козленко, Н.В. Кузь, И.Н. Лазарович, М.С. Дутчак

КВАНТОВЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ВЫСШИХ ПОРЯДКОВ
В ЗАДАЧАХ ФУНКЦИОНАЛЬНОЙ ОПТИМИЗАЦИИ

При построении квантовых генетических алгоритмов (QGA) традиционным является представление квантовой хромосомы в виде системы независимых кубитов. Это не позволяет использовать такой мощный механизм квантовых вычислений, как запутанность квантовых состояний. В работе реализован QGA высших порядков и проиллюстрировано его эффективность на примере задачи числовой оптимизации с использованием ряда тестовых функций. Также предложен оператор квантового гейта с адаптивным характером работы, не требующий использования таблицы поиска. В сравнении с традиционным QGA переход к высшим, более двух, порядкам при реализации алгоритма показывает значительно лучшие результаты как по времени работы, так и скорости сходимости и точности найденного решения.

К л ю ч е в ы е с л о в а: функциональная оптимизация, запутанность квантовых состояний, квантовый генетический алгоритм, квантовые вычисления, квантовый регистр.

TKACHUK Valerii Mykhailovych, Ph.D. (Phys.-Math.), Associate professor of the Vasyl Stefanyk Precarpathian National University, graduated from the Ivan Franko Lviv State University in 1984. The field of scientific interests: machine learning and data mining, artificial intelligence, quantum information, evolutionary algorithms.

KOZLENKO Mykola Ivanovych, Ph.D. (Eng.), Head of the Department of Information Technology and Associate Professor at Vasyl Stefanyk Precarpathian National University, graduated from Ivano-Frankivsk State Technical University of Oil and Gas in 1994. The field of scientific interests: robotics, deep learning for computer vision, software design and development.

KUZ Mykola Vasyliovych, Dr.Sc. (Tech.), Professor of the Vasyl Stefanyk Precarpathian National University, graduated from the Lviv Polytechnic National University in 1997. The field of scientific interests: software quality, evolutionary algorithms.

LAZAROVYCH Ihor Mykolaiovych, Ph.D. (Tech.), Associate Professor of the Vasyl Stefanyk Precarpathian National University, graduated from the Ivano-Frankivsk State Technical University of Oil and Gas in 2000. The field of scientific interests: artificial intelligence, machine learning, noise-immune data transmission, signal randomization, digital data processing.

DUTCHAK Mariia Stepanivna, Assistant of the Vasyl Stefanyk Precarpathian National University, which graduated in 2007. The field of scientific interests: adaptive knowledge transfer system, data mining, expert systems.