Министерство по развитию информационных технологий и коммуникаций Республики Узбекистан

Министерство инновационного развития Республики Узбекистан

Научно-исследовательский институт развития цифровых технологий и искусственного интеллекта

Самаркандский филиал Ташкентского университета информационных технологий имени Мухаммада ал-Хоразмий

СОВРЕМЕННОЕ СОСТОЯНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ЦИФРОВЫХ ТЕХНОЛОГИЙ И ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Сборник докладов республиканской научно-технической конференции

Самарканд
26-27 октября 2022 г.

Часть 2

ТАШКЕНТ - 2022

Министерство по развитию информационных технологий и коммуникаций
Республики Узбекистан

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ РАЗВИТИЯ ЦИФРОВЫХ ТЕХНОЛОГИЙ И ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

САМАРКАНДСКИЙ ФИЛИАЛ
ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ

СОВРЕМЕННОЕ СОСТОЯНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ЦИФРОВЫХ ТЕХНОЛОГИЙ И ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Сборник докладов республиканской научно-технической конференции

Самарканд, 26-27 октября 2022 г.

ЧАСТЬ 2

Ташкент – 2022

UDC 519.6

EFFECTIVE METHODS OF OPTIMIZATION OF QUANTUM ALGORITHMS

Toirov Sh.A., Boynazarov I.M., Kudratov R.B., Kholmatov O.A.

*tshuxrat@mail.ru
Samarkand branch of Tashkent University of Information Technology

Abstract. In the paper the quantum computation language used in solving quantum algorithms, its operators, and the principles of solving using the Deutsch algorithm are described. The basic principles, physical and algorithmic interpretations of quantum processes are taken into account. On the basis of these processes, the classification of the algorithm used in the analysis of the system in search of effective solutions to the problems of global optimization and rational management of unexpected situations is presented.

Keywords: Quantum, genetic and immune algorithms, qcl (quantum computation language), global optimization, intelligent control.

1 Introduction

The search for a solution to the problem of global (generally multi-dimensional) optimization is typical for systematic analysis. The uncertainty of information and the adoption of optimal solutions and management of complex systems in risky conditions have been evolving over the years in a variety of areas. In recent years, new forms of intelligent computing have been used to solve this problem.

To find a solution using a quantum algorithm, a series of quantum operators are used, changing the initial state according to the purpose of the initial superposition.

In traditional conventional programming, a one-parameter function is implemented as follows:

• the parameter is placed in the incoming register;

• The commands that make up the body of the function perform some manipulations (changes) on this parameter, and then the result is placed in the output register, where the previous state of the parameter is lost.

In quantum programming, the last operation is impossible because it is irreversible. Instead, the resulting bits are added to the result of dividing the output register *(⨁)* by 2 modules. In other words, the *XOR* (exceptional or) operation is performed on them. This process is obviously reversible: it is enough to use it a second time and the memory returns to its original state [1].

2 The main part

Quantum parallelism is one of the key features of quantum computing. This feature allows quantum computers to simultaneously calculate the function *f(x)* for different values of *f*. To describe quantum parallelism, we consider the calculation of the function of the variable *x*, which is described in the following figure.

$$f(x)\ :\ \{0,\ 1\}\ \rightarrow\ \{0,\ 1\}$$

In a quantum computer, the optimal way to calculate this function is to consider a two-qubit quantum computer, and it works with the |xy> state. By applying a sequence of logical gates, you can change the initial $|\,xy > state\ to\ the\,|\,x,\ y \otimes f(x) > state$. Here we can say that *x*, *y* are the registers of a quantum computer. In this case, the first register is called **the data register**, and the second is

the target register. Here the variables $|x,\ y>\to|x,\ y\otimes f(x)>$ are represented by the unitary variable *U*.

Each of them is associated with a unitary operator. That is, $U_f\,|\,x,y> \ = \ |\ x,\ y\oplus f(x)>$.

In such cases, the function will be in four cases: that is, in cases where $f(x)=0$ and $f(x)\ =1$, the function $f(x)$ is constant, in cases where $f(x)=x$ and $f(x)=1\ -\ x$, the function will be balanced (since it takes the values 0 and 1 in points with equal numbers) [4].

In solving this task $f(0)\otimes f(1)$ is regarded as a multiplex on the two modules. To calculate this process $|x\rangle\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$ (1) it is enough to find solutions in the case where is $f(0)=0$ and $f(1)=0$. In that case, if $x=0$ and $x=1$, then the appearance of (1) is as follows $\frac{(|00\rangle-|01\rangle)}{\sqrt{2}}$ and $\frac{(|10\rangle-|11\rangle)}{\sqrt{2}}$ will be.

If $f(0)=1$ and $f(1)=1$, then in the case where $x=0$ and $x=1$, then the appearance of (1) is as follows $|-1\rangle\frac{|00\rangle-|01\rangle}{\sqrt{2}}$ and $|-1\rangle\frac{|10\rangle+|11\rangle}{\sqrt{2}}$ will be.

The above two cases can be written as follows through the U_f unitary variable

$$U_f\left[|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]=(-1)^{f(x)}|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} \tag{2}$$

Here $|-\rangle\ =\ \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ we enter the sign. Now let's look at the initial case, namely $|+>\ *\ |->$. Then $|\psi\rangle=\frac{|0\rangle}{2}|-\rangle+\frac{|1\rangle}{2}|-\rangle$ will be equal. Now if we consider the $UF\ /\ \psi 1>$ unitary variable, it will be as follows. [3]

$$U_f|\psi_1\rangle=(-1)^{f(0)}\frac{|0\rangle}{2}|-\rangle+(-1)^{f(1)}\frac{|1\rangle}{2}\,|-> \tag{3}$$

(3) from the formula it can be seen that if $f(x)$ the function will be constant, when $(-1)^{f(1)}=(-1)^{f(0)}$ and the result $(-1)^{f(0)}|+\ >\otimes|->$ when there is. If $f(x)$ is a function balanced, when $(-1)^{f(1)}=-(-1)^{f(0)}$, and the result is $(-1)^{f(0)}\,|->\otimes|->$ when [2].

As can be seen from the above, when you apply M to the first bit. For a constant function $|\,0>$ and for a balanced function $|\,1>$ is.

Now, let's look at the functionality we've discussed above using the QCL Quantum Computing Language programming language.

Required descriptions:

```
qcl> qureg x[1]; qureg y[1]; int  r;
```

We write out the operator U_F:

- for n = 0- f(x) = 0 (this operator does not perform anything);
- for N = 1 – let f(x) = 1;
- for N = 2 – let f(x) = x;
- for N = 3 – let f(x) = 1 – x.

Now let's consider the following process in QCL Quantum Computing Language Programming Language [2].

```
qcl> procedure U(int  n, qureg  x, qureg  y) { if n==1 { Not(y); }  /* f(x)=1 */
else { if n==2 { x->y; }  /* f(x)=x */ else { if n==3 { Not(x); x->y; Not(x); }}}
/* f(x)=1-x */}
```

After we enter this process, we will consider the following case of the Y bit / > y bit / > y:

```
qcl> Not(y)
```

```
[2/32] 1 |0, 1>
qcl> Mix(y)
[2/32] 0.70711 |0, 0> – 0.70711 |0, 1>
```

Now $|+> \otimes / - >$ when reviewing the status gets the following view of its result.

```
qcl> Mix(x)
[2/32] 0.5 |0, 0> + 0.5 |1, 0> – 0.5 |0, 1> – 0.5 |1, 1>
```

Now we see that the appearance of the U_f operator when the U_f operator corresponding to the function *f(x)=1* is used is as follows. The result is shown in the lower row.

```
qcl> U(1,x,y)
[2/32] –0.5 |0, 0> – 0.5 |1, 0> + 0.5 |0, 1> + 0.5 |1, 1>
```

Proceeding from the above, X will have the following appearance when using the bit-to-bit Adamar function.

```
qcl> Mix(x)
[2/32] –0.70711 |0, 0> + 0.70711 |0, 1>
qcl> measure x,r
[2/32] –0.70711 |0, 0> + 0.70711 |0, 1>
qcl> print r
0
```

And we get a state in which the result is zero. It can be seen that in the case of *x* bits *|0>*, of course, the function will be constant. *x* bits write and print the result of the value to a variable *r* [2].

Now we will bring the memory back to its original state to consider how much the result of other functions will also be, and for another function of *f(x)*, we will repeat the above cases when *f(x)=x*. Here is the operator to *reset* memory to its original state.

```
qcl> reset
[2/32] 1 |0, 0>
qcl> Not(y)
[2/32] 1 |0, 1>
qcl> Mix(y)
 [2/32] 0.70711 |0, 0> – 0.70711 |0, 1>
qcl> Mix(x)
[2/32] 0.5 |0, 0> + 0.5 |1, 0> – 0.5 |0, 1> – 0.5 |1, 1>
qcl> U(2,x,y)
[2/32] 0.5 |0, 0> – 0.5 |1, 0> – 0.5 |0, 1> + 0.5 |1, 1>
qcl> Mix(x)
[2/32] 0.70711 |1, 0> – 0.70711 |1, 1>
qcl> measure x,r
[2/32] 0.70711 |1, 0> – 0.70711 |1, 1>
qcl> print r
1
```

even if *f(x)=1-x*, the result is a state equal to 1.

```
qcl> reset
[2/32] 1 |0,0>
qcl> Not(y)
[2/32] 1 |0,1>
qcl> Mix(y)
[2/32] 0.70711 |0,0> - 0.70711 |0,1>
qcl> Mix(x)
[2/32] 0.5 |0,0> + 0.5 |1,0> - 0.5 |0,1> - 0.5 |1,1>
qcl> U(3,x,y)
[2/32] -0.5 |0,0> + 0.5 |1,0> + 0.5 |0,1> - 0.5 |1,1>
qcl> Mix(x)
[2/32] -0.70711 |1,0> + 0.70711 |1,1>
qcl> measure x,r
[2/32] -0.70711 |1,0> + 0.70711 |1,1>
qcl> print r
1
```

As can be seen from the result obtained, *x* bits are equal to 1, then of course *f(x) = x* and *f(x) =1 - x* functions will be balanced [1].

Next, we can write a process in which the whole algorithm of work is automated. In the process, the parameter *n* uses the function *f(x)*.

```
qcl> procedure Deutsch( int n)
{ reset; Not(y); Mix(y); Mix(x); /* |+> * |-> */  U(n,x,y); Mix(x);
measure x,r; print r; }
qcl> Deutsch(0)
0
[2/32] 0.70711 |0, 0> – 0.70711 |0, 1>
qcl> Deutsch(1)
0
[2/32] –0.70711 |0, 0> + 0.70711 |0, 1>
qcl> Deutsch(2)
1
[2/32] 0.70711 |1, 0> – 0.70711 |1, 1>
qcl> Deutsch(3)
1
[2/32] –0.70711 |1, 0> + 0.70711 |1, 1>
qcl> exit
```

We see from this that when the algorithm is actually equal to 0, the first two $f(x)$ functions will be balanced for the remaining two $f(x)=x$ and $f(x)=1-x$ when the constant $(f(x)=0, f(x)=1)$ and the remaining two $(f(x) = x$ and $f(x) = 1-x)$ [2].

3 Conclusion

In summary, in this article, the qcl (quantum computing language) language used in solving quantum algorithms, its operator's processing processes and Deutsch's algorithm processing principles are discussed. The basic principles of quantum processes, physical and algorithmic interpretations were taken into account. On the basis of these processes, the classification of the algorithm used in the analysis of the system in search of effective solutions to the problems of global optimization and rational management of unexpected situations is presented.

Unlike the classical analog, quantum algorithm yacheyka can be performed in different classes of universal elements, depending on the basis of the calculation used. Quantum algorithm slope describes the evolution of some unitary operator *U*, corresponding to the process of quantum computing.

References

[1] *Ulyanov S. V.* Quantum fast algorithm computational intelligence PT I; SW, HW smart toolkit// Artificial Intelligence Advances. 2019. -№1. pp.18-43.

[2] *Ulyanov S.V., Litvintseva L.V., Ulyanov S.S.* Quantum information and quantum computational intelligence: Quantum optimal control and quantum filtering Stability, robustness, and self-organization models in nanotechnologies // Universita degli Studia di Milano. 2005. – Vol. 82. – pp.109-124.

[3] *Ulyanov S.V., Litvintseva L.V., Ulyanov S.S.* Quantum information and quantum computational intelligence: Applied quantum soft computing in AI, quantum language and programming in computer science, quantum knowledge self-organization and intelligent wise robust control (4rd edit.)// Universita degli Studia di Milano. – 2010. – Vol. 86. – pp.279-301.

[4] *Ohwi J., Ulyanov S.V., Yamafuji K.* GA in continuous space and fuzzy classifier system for opening a door with a manipulator of mobile robot: New Benchmark of evolutionary intelligent computing // Journal of Robotics and Mechatronics, 1996. – Vol.8. – № 3. – pp. 297-301.

[5] *Ulyanov S.V., Litvintseva L.V., Takahashi K.* Fast algorithm for efficient simulation of quantum algo- rithm gates on classical computer // Systemics, Cybernetics and Informatics. – 2004. – Vol. 2. – №3. – pp. 63-68.

[6] *Литвинцева Л.В., Ульянов С.В.* Интеллектуальные системы управления. I. Квантовые вычисления и алгоритм самоорганизации // Изв. РАН. ТиСУ. 2009. № 6. С. 69–97.

[7] *Ulyanov S.V., Litvintseva L.V., Ulyanov S.S.* Quantum Information and Quantum Computational Intel- ligence: Design & Classical Simulation of Quantum Algorithm Gates// Universita degli Studia di Milano. – 2000. – Vol. 80. – pp.60-68.

[8] *Ulyanov S.V., Litvintseva L.V., Ulyanov S.S.* Quantum information and quantum computational intelligence: Quantum probability, physics of quantum information and information geometry, quantum computational logic and quantum complexity// Universita degli Studia di Milano. – 2003. – Vol. 80. – Pp.80-82.

[9] *В. С. Потапов, С. Гушанский* Определение и реализатсия операторов квантовых алгоритмов// Жувенис ссиентиа, 2016. № 2. пп. 38-40.

[10] *С. Ульянов, А. Полунин* Моделирование на классическом компьютере квантового алгоритма принятия решения дейча// Электронный журнал «Системный анализ в науке и образовании», 2010. № 2. пп. 76-89.

[11] *Deutsch D.* Quantum Theory the Church-Turing Principle and the Universal Quantum Computer// London, 2006. V A400.1818. pp. 23-31.

[12] *Deutsch D., Jozsa R.* Rapid Solution of Problems by Quantum Computation// Proc. Roy. Soc. London, 2008. V A439. 1907. pp.141-158.

[13] Quantum Algorithms: Shor's algorithm, Grover's algorithm, Quantum logic, Quantum algorithm, Quantum Fourier transform, Deutsch-Jozsa algorithms// – Books LLC. – 2010. pp.53-74.

[14] *Chen G., Kauffman L., and Lomonaco S. J.* Mathematics of Quantum Computation and Quantum Technology//Applied Mathematics and Nonlinear Science Series. – 2008. pp.421-484.

[15] *Toirov Sh.* Efficient methods for solving functions using quantum genetic algorithms//XI International Scientific Conference “Theoretical approaches of Fundamental Sciences. Theory, Practice and prospects", Geneva, Switzerland, 2021. p. 45-50.

СОДЕРЖАНИЕ