УДК 530.145.6

KVANT ALGORITMLARNI OPTIMALLASHTIRISHNING SAMARALI YECHISH USULLARI

Toirov Sh. A.

Ushbu maqolada kvant algoritmlarni yechishda qo‘llaniladigan qcl (quantum computation language), uning operatorari va Deutsch algorimini orqali yechish prinsiplari keltirib o‘tilgan. Kvant jarayonlarning asosiy tamoyillari, fizikaviy va algoritmik talqinlari hisobga olingan. Ushbu jarayonlar tizimni tahlil qilishda global optimallash muammolariga samarali yechimlarni qidirishda va kutilmagan holatlarni oqilona boshqarishda qo‘llaniladigan algoritmning tasniflari keltirilib o‘tilgan.

Tayanch iboralar: Kvant, genetik va immun algoritmlar, qcl (quantum computation language), global optimallashtirish, intellektual boshqarish.

В данной статье рассматриваются принципы решения языка квантовых вычислений, его оператор и немецкий алгоритм, используемый при решении квантовых алгоритмов. Учтены основные принципы квантовых процессов, физические и алгоритмические интерпретации. Эти процессы описывают классификацию алгоритмов системы, которая используется для поиска эффективных решений глобальных задач оптимизации и управления непредвиденными ситуациями.

Ключевые слова: квантовые, генетические и иммунные алгоритмы, qcl (quantum computation language), глобальная оптимизация, интеллектуальное управление.

This article deals with the principles of solving quantum computing language, its operator and the German algorithm used in solving quantum algorithms. The basic principles of quantum processes, physical and algorithmic interpretations are taken into account. These processes describe the system's algorithm classification, which is used to search effective solutions to global optimization problems and to manage the unexpected situations.

The search for a solution of the Global (Multi-phase Multiple) optimization problem is typical for system analysis. The uncertainty of information and the adoption of optimal solutions in risky conditions and the management of complex systems have been advancing for years. In recent years, the solution to this problem has been successful with new ideas of intellectual computing.

In order to find a solution with the help of quantum algorithm, quantum operators are used in the same sequence as changing the initial state to the initial superposition.

In quantum programming, the last operation (operation) is impossible, because this is not permitted. Instead, the results are added to the result of splitting out the output log (⊕) into two modules. In other words, the XOR (exception) action is performed on them. This action, of course, is reversible: it is enough to use it again and the memory returns to its original state.

Unlike the classic analogue, the algorithm of the cavern can be executed in different classes of universal elements, depending on the basis of the calculation used. The cube-algorithm cell describes the evolution of some unitary operator, which corresponds to the quantum process.

Keywords: quantum, genetic and immune algorithms, qcl (quantum computation language), global optimization, intellectual management.

I. KIRISH

Global (umumiy holda ko‘pmezonli) optimallash masalasining yechimini qidirish tizimli tahlil uchun odatiy hisoblanadi. Axborotning noaniqligi va xatarli (riskli) shartlarda optimal yechimlarni qabul qilish va murakkab tizimlarni boshqarish har xil yo‘nalishlarda ko‘p yillardan buyon rivojlanib kelmoqda. So‘nggi yillarda mazkur masalaning yechimi intellektual hisoblashlarning yangi ko‘rinishlari bilan muvaffaqiyatli topilmoqda.

Kvant algoritmning yordami bilan yyechimni topish uchun, daslabki xolatni dastlabki superpozitsiya maqsadiga muovofiq ravishda o‘zgartirib, kvant operatorlarini ketma-ket ko‘rsatilgan turlari qo‘llaniladi.

Odatdagi an’anaviy dasturlashda bir parametrli funksiya quyidagicha amalga oshiriladi:

- parametr kiruvchi registrga joylashtiriladi;
- funksiya tanasini tashkil etuvchi komandalar bu parametr ustida biror manipulyasiyalar (o‘zgartirishlar) amalga oshiradi, so‘ngra natija chiqish registriga joylashtiriladi, bunda parametrning ilgarigi holati yo‘qotiladi.

Kvantli dasturlashda esa so‘nggi amal (operatsiya) imkonsiz, chunki bu amal qaytarilmasdir. Buning o‘rniga natijalar bitlari chiquvchi registrning ($\oplus$) 2 modul bo‘yicha bo‘linishi natijasiga qo‘shiladi. Boshqacha aytganda, ular ustida *XOR* (istisnoli yoki) amali bajariladi. Bu amal, aniqki, qaytariluvchi: uni ikkinchi marta qo‘llash etarli va xotira boshlang‘ich holatiga qaytadi [1].

II. АСОСИЙ ҚИСМ

Masalani qo’y ilish va yechimlari

Kvant parallelizmi kvant hisoblashning asosiy xususiyatlaridan biri hisoblanadi. Bu xususiyat kvantli kompyuterlarga f ning turli qiymatlari uchun $f(x)$ funksiyasini bir vaqtning o'zida hisoblash imkonini beradi. Kvant parallelizmini tasvirlash uchun x o'zgaruvchisining funksiyasini hisoblashni ko'rib chiqamiz va u qo‘yidagi ko‘rinishda tasvirlanadi.

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

Kvant kompyuterda bu funksiyani hisoblashning maqbul usuli - ikki qubitli kvantli kompyuterni ko'rib chiqishdan iborat va u $|xy>$ xolat bilan ishlaydi. Mantiqiy eshiklar ketma-ketligini qo'llash orqali, boshlang'ich $|xy>$ holatni $|x, y \otimes f(x)>$ xolatga o'zgartirishi mumkin. Bu erda x, y - kvant kompyuterning registrlari deb ayta olamiz. Bunday holatda, birinchi regestr **ma'lumotlar regestri** deb ataladi va ikkinchisi - **nishon (metka) regestr**. Bu erda $|x, y> \rightarrow |x, y \otimes f(x)>$ o'zgaruvchilarni U unitar o‘zgaruvchi orqali amalga oshiriladi.

Ularning har biri unitar operator bilan bog'liq. Yani $U_f|x, y> = |x, y \oplus f(x)>$.

Bunday xolatlarda funksiya to‘rtta xolatda bo’ladi: ya’ni $f(x) = 0$ va $f(x) = 1$ bo‘lgan xolatlarda $f(x)$ funksiya doimiy, $f(x) = x$ i $f(x) = 1 - x$ bo‘lgan xolatlarda $f(x)$ funksiya muvozanatlashgan bo‘ladi (chunki 0 va 1 qiymatlarini teng sonli nuqtalarda oladi) [4].

Bu masalani echishda $f(0) \otimes f(1)$ ikki modul bo‘yicha ko‘paytmasi qaraladi. Bu jarayonni hisoblash uchun (1) ni $f(0)=0$ va $f(1)=0$ bo‘lgan holatda echiimlarini topish etarlidir. U xolda, agar $x=0$ va $x=1$ bo‘lganda (1) ning ko‘rinishi qo‘yidagicha (|00> −|01>)2 va (|10> bo‘ladi.

Agar $f(0)=1$ va $f(1)=1$ bo‘lgan holatda esa, $x=0$ va $x=1$ bo‘lganda (1) ning ko‘rinishi qo‘yidagicha bo‘ladi.
yuqoridagi ikki xolatni U_f unitar o‘zgaruvchi orqali qo‘yidagicha yozish mumkin

$$Uf|x>|0>-|1>2=-1fx|x>|0> -|1>2 \quad (2)$$

bu erda > belgilashni kritamiz. Endi dastlabki xolatni ko‘rib chiqamiz ya’ni $|+>$ * $|->$. U holda > teng bo’ladi. Endi $U_f|\psi_1>$ unitar o’zgaruvchini hisoblasak, u quyidagicha bo’ladi. [3]

$$Uf|\psi 1> = -1f0|0>2 \ |-> + -1f1|1>2 \ |-> \quad (3)$$

(3) formuladan ko‘rinib turibdiki, agar $f(x)$ funksiya doimiy bo‘ladi, qachonki $(-1)^{f(1)} = (-1)^{f(0)}$ bo‘lsa, va natija $(-1)^{f(0)} |+> \otimes |->$ bo‘lganda. Agar $f(x)$ funksiya muvozanatlashgan bo‘ladi, qachonki $(-1)^{f(1)} = -(-1)^{f(0)}$ bo‘lsa, va natija $(-1)^{f(0)} |-> \otimes |->$ bo‘lganda [2].

Yuqoridagilardan ko‘rinib turibdiki M ni birinchi bitga qo'llaganda. Doimiy funksich uchun $|0>$ va muvozanatlashgan funksiya uchun $|1>$ bo‘ladi.

Endi yuqorida kurib chiqqan funksiyani QCL Quantum Computation Language dasturlash tili yordamida ko‘rib chiqamiz.

Kerakli tavsiflar:

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

U_f operatorini yozib chiqamiz:

- $n = 0$ – uchun $f(x) = 0$ (bu operator xech narsa bajarmaydi);
- $n = 1$ – uchun $f(x) = 1$ bo’lsin;
- $n = 2$ – uchun $f(x) = x$ bo’lsin;
- $n = 3$ – uchun $f(x) = 1 - x$ bo’lsin.

Endi QCL Quantum Computation Language dasturlash tilida ko‘yidagi jarayonni yozib chiqamiz [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 */}

Bu jarayonni kritganimizdan so‘ng, y bitga $|-\rangle$ xolatni ko‘rib chiqamiz va quyidagi natijaga ega bo‘ladi:

qcl> Not(y)
[2/32] 1 |0, 1>
qcl> Mix(y)
[2/32] 0.70711 |0, 0> − 0.70711 |0, 1>

Endi $|+\rangle \otimes |-\rangle$ xolat ko‘rib chiqilganda uning natijaga qo‘yidagi ko‘rinishni oladi.

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

Endi $f(x) = 1$ funksiyaga mas keladigan U_f operator qo‘llanilganda U_f operatorining ko‘rinishi qo‘yidagicha bo‘lishini ko‘ramiz. Natijasi pastki qatorda ko‘rsatilgan.

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

Yuqoridagilardan kelib chiqib, x bitga Adamar funksiyani qo‘laganda qo‘yidagi kurinshga ega buladi.

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

Va natija nolga teng bo‘lgan xolatni olamiz. Bundan ko‘rinadiki, x bit $|0\rangle$ xolatda albatta funksiya doimiy bo‘ladi. x bitni qiymat natijasini o‘zgarmaydigan r ga yoziladi va chop etadi [2].

Endi boshqa funksiyalarni xam natijasi qanaqa bulishligini ko‘rib chiqish uchun xotirani asl xoliga keltiramiz va $f(x)$ ning yana bitta funksiyasi uchun yuqoridagi xolatlarni takrorlaymiz $f(x)=x$ bo’lganda. Bu erada *reset* xotirani asl xoliga keltirish operatori.

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
```

f (x)=1-x bo’lganda xam natijasi 1 ga teng xolat chiqayapdi.

```
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
```

Olingan natijadan ko‘rinib turibdiki, *x* bit 1 ga teng bo‘lyapdi, u xolda albatta *f(x) = x* va *f(x) =1- x* funksiyalar muvozanatlashgan bo‘ladi [1].

Keyingi ish butun algoritmni avtomatlashagan jarayonni yozishimiz mumkin. Bu jarayonda *n* parametr *f(x)* funksiyani ishlatadi.

```
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
```

Bundan biz ko‘ramizki, algoritm xaqiqatdan xam 0 ga teng bo‘lganda birinchi ikkita *f(x)*funksiyalar doimiy (*f(x)= 0, f(x)=1*) va 1 ga ten bo‘lganda qolgan ikkita (*f(x)=x* va *f(x)= 1-x*) uchun funksiya muvozanatlashgan bo‘ladi [2].

III. XULOSA

Xulosa qilib aytadigan bulsak ushbu maqolada kvant algoritmlarni echishda qo‘llaniladigan qcl (quantum computation language) tili, uning operatorarining ishlash jarayonlari va Deutsch algorimini ishlash prinsiplari keltirib o‘tilgan. kvant jarayonlarning asosiy tamoyillari, fizikaviy va algoritmik talqinlari hisobga olingan. Ushbu jarayonlar tizimni tahlil qilishda global optimallash muammolariga samarali yechimlarni qidirishda va kutilmagan holatlarni oqilona boshqarishda qo‘llaniladigan algoritmning tasniflari keltirilib o‘tilgan.

Klassik analogdan farqli o‘laroq, kavnt algoritm yacheyka ishlatiladigan hisoblash asosiga qarab universal elementlarning turli sinflarida bajarilishi mumkin. Kavnt algoritm yacheykasi kvant hisoblash jarayoniga mos keladigan ba’zi bir unitar operator *U* ning evolyutsiyasini tavsiflaydi.

ADABIYOTLAR

[1] B. Omer, Structured Quantum Programming, Ph. D. Thesis, Technical University of Vienna (2003); http://tph.tuwien.ac.at/~oemer/

[2] А. Г. Грозин, Квантовый компьютер для чайников, Новосибирск(2004).

[3] J. van der Hoeven, GNU TEXmacs, http://www.texmacs.org/

[4] A.G. Grozin, TEXmacs interfaces to Maxima, MuPAD and REDUCE, Proc. 5 Int. workshop on Computer algebra and its applications to physics, Dubna, JINR E5,11-2001-279; cs.SC/0107036.

[5] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computer, Proc. Roy. Soc. A 439 (1992) 553.

[6] P.W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, Proc. 35 Annual Symposium on Foundations of Computer Science, IEEE Press (1994).

[7] *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Computing 26 (1997) 1484.*

[8] R. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Comm. ACM 21 (2) (1978) 120.

[9] C.H. Bennett, G. Brassard, and N.M. Mermin, Quantum cryptography without Bell's theorem, Phys. Rev. Lett. 68 (1992) 557.

[10] S.Wiesner, Conjugate coding, написано в 1970, опубликовано в SIGACT News 15 (1) (1983) 78.

[11] C.H. Bennett, G. Brassard, C. Cr´epeau, R. Jozsa, A. Peres, and W.K. Wootters, Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, Phys. Rev. Lett. 70 (1993) 1895.