

Analysis of Quantum Algorithms with Classical Systems Counterpart

Shyam R. Sihare

Asstt. Professor, Department of Computer Science and Application,
 Dr. APJ Abdul Kalam Govt. College, Silvassa, Dadra & Nagar Haveli (UT)-396235, India
 Email: shyams_sihare1979@rediffmail.com

Dr. V V Nath

Professor, Institute of Management, Nirma University, Ahmedabad, Gujarat, India
 Email: vvnath01@gmail.com

Abstract—In this note, we look into two quantum algorithms, Deutsch-Josza’s and Shor’s algorithms. An attempt made to analyze classical as well as quantum parts computation. With that, analyze classical as well as quantum parts complexities.

Index Terms—BQP(Bounded-error Quantum Polynomial time), BPP(Bounded-error Probabilistic Polynomial time), Quantum Algorithms, Classical Algorithms, Deutsch-Jozsa.

I. INTRODUCTION AND RELATED WORK

Due to speed up feature of quantum computer [1] compelled to shape it into commercial viability. Nevertheless, still it is in an infancy stage. During successive time number of algorithms has been shaped before quantum computer come into a workable form such as Deutsch algorithm [2], Deutsch-Josza algorithm [3], Simon’s algorithm [4], Grover algorithm [5], Shor’s algorithm [6] and more. Most of these algorithms attempted to solve (theoretically or partial experimentally) the classical Turing machine NP problems in polynomial time $p(x)$. Nowadays high-speed digital computers even take exponential time for BPP problems. It is hard to imagining that solving NP problems in polynomial time $p(x)$ by classical Turing machine. The Deutsch algorithm was the first quantum algorithm try to solve constant and balanced qubits [2] in $p(x)$.

There are numerous unsolvable problems incorporated into NP. As we said earlier that, it is quite impossible to solve NP problems by Deterministic and probabilistic Turing machine in polynomial time $p(x)$.

Algorithms computation performance measures by complexities classes in digital systems. Furthermore, complexities classes divided into computation and storage space performance. In a deterministic Turing machine, the probability of solving these problems into polynomial time is out of the compass by P=? NP[7]. It turns out that every NP problems are not solvable under the influence of quantum algorithms interpretation. Some extend NP problems of deterministic Turing machine

could be solve through a quantum computer by using the parallelism properties (*i.e.* superposition), entanglement and qubits measurement frequency.

II. BPP VS BQP

BPP stands for Bounded-error Probabilistic Polynomial time (*i.e.* decision problems solvable by probabilistic Turing machine in polynomial time with certain bounded error, say arbitrary Q).

BQP is complexity class of quantum algorithms. It stands as “Bounded Error, Quantum, and Polynomial time”. Means, it included those quantum algorithms which solvable by polynomial time $p(x)$, x is a number of instructions in time t . BQP classify as (1) Bounded error (*i.e.* input x error confine lower bound $\frac{1}{3}$. Similarly upper bound confine $\frac{1}{4}$). (2) Quantum—quantum algorithm x with properties of entanglement and superposition. An entanglement has had two states $|\psi\rangle$ and $|\phi\rangle$ at two different times t_1 and t_2 respectively are related with each others.

$$|\Psi\rangle = U|\phi\rangle \quad (1)$$

Where $UU^\dagger = U^\dagger U = I$, U is unitary matrix, U^\dagger is adjoint matrix of $U^\dagger = (U^T)^*$, I identity matrix [1].

Evidently, entanglement of $|r_i\rangle$, $0 \leq i \leq n$, in form of

$$|r_0\rangle|r_1\rangle\dots|r_n\rangle, n \in I^+ \quad (2)$$

Thus,

$$|r_i\rangle = \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix} \otimes \begin{vmatrix} 0 & 1 \\ 0 & 0 \end{vmatrix} \dots \quad (3)$$

That is, it reflects multiple places at the same time for computation and collapse at the input/output measurement time as bases states $|0\rangle$ or $|1\rangle$. (3) Polynomial time $p(x)$ —quantum algorithms solvable in polynomial time $p(x)$, but as per Church-Turing thesis, this computation intuitive contrary.

Hence, all decision problems find “yes” or “no” answers with an error probability of $\frac{1}{3} \leq x \leq \frac{1}{4}$, where x is polynomial solvable decision problems.

Further, $BQP^{BQP} = BQP$, i.e. polynomial time algorithms calls a polynomial subroutine one after another with particular sequences, on consequence the resulting algorithms is still polynomial time after completing all processing [8]. Expansion of it as

$$BQP^{n_1^{n_2 \dots n_i}} = BQP, 1 \leq i \leq p, n_i = BQP \quad (4)$$

Eq. (4) depends on a number of qubits of a quantum computer. The deep subroutine of quantum algorithms needs exponential qubits such as $[n_1, [n_2, [n_3, [\dots, n_i]]]$, nested subroutine need asymptotic $O(n^2)$ qubits for computation which violate quantum polynomial phenomena. This is an open question for arguments.

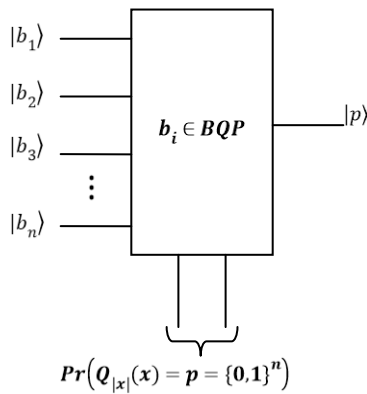


Fig.1. Quantum Turing machine solve all classical unsolvable problems into BQP time

At input end of quantum Turing machine possesses n -number of qubits which is state in fig.1. Here quantum Turing machine is capable to solve all BQP problems. All $|b_n\rangle$ qubits feed into BQP quantum Turing machine and generate intermediate result as $Pr(Q|x)(x) = p = \{0,1\}^n$. Quantum state Q with function x solve into polynomial time p on n^{th} binary input. After completion of intermediate result, generate $|p\rangle$ result at output end (fig. 1).

Possible difference between different classes is $P \subseteq BPP \subseteq BQP \subseteq AWPP \subseteq PP \subseteq PSPACE$, however difficult to solve P problems in $PSPACE$ i.e. $P \subseteq ?PSPACE$ [9].

Thus, $P \subseteq BQP$ indicate that quantum circuit simulate classical circuit as well as simulate $BPP \subseteq BQP$ problems; quantum computer can solve those problems which solved by probabilistic computer. Thus quantum property gives us randomness [10]. The $BPP \subseteq EXP$ problems as in quantum state $|\psi\rangle = \alpha_1|x\rangle + \alpha_2|x\rangle + \dots + \alpha_i|x\rangle$ of quantum computer solve all BPP problems within exponential time.

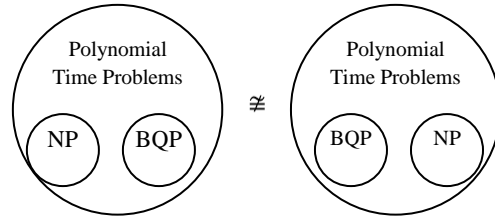


Fig.2. Congruency between $NP \not\subseteq BQP$ and $BQP \not\subseteq NP$

The $NP \not\subseteq BQP$ problems means NP problems does not contain into BQP. In fact, it is impossible to solve all NP problems through BQP. Contrary to it, $BQP \not\subseteq NP$ indicate that BQP problems solvable as NP problems minimally but future wait maximally i.e. unpredictable till time.

III. DEUTSCH-JOZSA ALGORITHM ANALYSIS

Before Deutsch-Josza algorithm, Deutsch’s algorithm concept took ground in the quantum computer world. Deutsch’s algorithm based on the constant and balanced function f

$$f = \{0,1\}^n \rightarrow \{0,1\} \quad (5)$$

Eq. (5) interpreted as $f(0)=0, f(0)=1, f(1)=0, f(1)=1$, means either it is constant or balanced. This is single input vector basis state $|x\rangle$ and its output also single vector which is $f(x)|x\rangle$ for $x \in \{0, 1\}$, either constant or balanced but not both at the same time even if quantum mechanics property superposition on state vector space applicable. But its time complexity directly proportional to the n -input qubits vector in an oracle box. Furthermore, application of Hadamard transforms H takes speedup on superposition of input vector [11]

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^1 (-1)^{xy} |y\rangle, x \in \{0,1\} \quad (6)$$

Where $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ and two qubits state become

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Deutsch’s algorithm operated on 1-dimensional vector space with single qubit input, contrary to this Deutsch-Josza’s algorithm work on n -qubits input i.e. $N=2^n$ dimensional Hilbert space, where n is n -qubits. For constant and balanced function f , all n -qubits input become $f(x)=0$ or $f(x)=1$. Next, for balanced function f , half of the n -input qubits as $f(x)=0$ exactly and rest is $f(x)=1$.

Hence, n -Hadamard transform ($H^{(n)}$) is applicable for n -qubits input [11].

$$H^{(n)} = H \otimes_1 H \otimes_2 H \otimes_3 \dots \otimes_n H \quad (7)$$

on state $|x\rangle$ is given by –

$$H^{(n)}|x\rangle = \prod_{i=1}^n \frac{1}{\sqrt{2}} \sum_{y_i=0}^1 (-1)^{x_i y_i} |y_i\rangle, x_i \in \{0,1\} \quad (8)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \quad (9)$$

, where $x \cdot y = \bigoplus_{i=1}^n x_i y_i$ is the scalar product of 2

Eq. (8) indicates that, application of n -Hadamard transform on $|x\rangle$, its output depending on the phase of $x_i \cdot y_i$.

In Deutsch-Jozsa's algorithm, $H^{(n)}$ (n -qubits Hadamard transform) and H (Hadamard transform) is applied on vector state $|0\rangle$ and $|1\rangle$ as tensor product respectively by which basis vector state $|x\rangle$ generated in 2^n-1 times and basis vector state $|0\rangle$ and $|1\rangle$ generated as a single qubit as an output. For changing vector state position into the Hilbert space then unitary transformation U_f is performed on the basis vector states namely $|0\rangle$ and $|1\rangle$. Further, apply Hadamard transform H with U_f for generating constant or balanced function $f(x)$ [12]

$$\left| {}^n\langle 0 | H U_f H^{(n)} | 0 \rangle \right|^2 = \left| (-1)^{f(x)} \right|^2 = 1 \quad (10)$$

and

$$\left| {}^n\langle 0 | H U_f H^{(n)} | 0 \rangle \right|^2 = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2 = 0 \quad (11)$$

IV. COMPUTATION COMPLEXITY OF DEUTSCH-JOZSA ALGORITHM AND CLASSICAL ALGORITHMS COUNTERPART

Deutsch-Jozsa algorithm claims resource utilization by uses of quantum computer parallelism. It uses the superposition principle of quantum mechanics for computation $f(x)$, whether it is constant or balanced. However, practical application was not prime objective, but quantum computation in polynomial time was prime objective which creates interest for the development of a number of quantum algorithms. On the basis of this algorithm further enhancement apparently appreciable into Shor's, Grover's, and Fourier's transform algorithms. The classical computer also simulates the Deutsch-Jozsa algorithm by corresponding parallelism properties of a classical computer but resource utilization into a classical computer is not appreciable such as computation time, space complexity since each entity having a separate computation for constant or balanced function. In a quantum computer, an oracle box which is in superposition form took input as a single query while classical computer requires more than one queries, minimum two queries require for checking function. However, time complexity does not enhance exponentially even n input qubits in this quantum

algorithm. In worst case, $2^{n-1}+1$ time required for constant or balanced $f(x)$ evaluation. In fact, classical computer required double computation compared to a quantum computer for this. Half calculation of $f(x)$ required for determining constant or balanced vector state $|0\rangle$ or $|1\rangle$ then need 2^{n-1} for rest of computation.

V. SHOR'S ALGORITHMS ANALYSIS

Shor's algorithm is a well-known algorithm in a quantum family. Once quantum computer commercially viable then it might be possible that Shor's algorithm practically verifiable on it. Due to Shor's algorithm world focus towards the quantum computer development. Generally, a quantum computer looks as speedy by application of superposition, entanglement, and parallelism. Superposition used in Shor's algorithm for factorizing large integer number within a fraction of time. The Shor's algorithm brief summary state as

- Select prime number N which non-negative integer.
- Determine period r of a sequence x i.e. $r = x^n \bmod N$, N is an integer number.
- Apply Fourier transform to determine factorize terms.
- Factorize with the help of period r and Fourier transform.

Shor's algorithm computation divided into classical computation as well as quantum computation. If factorize integer number completely by quantum way then we cannot achieve polynomial performance [13]

a. Classical Computation Parts of Shor's Algorithm

1. Determine n , whether it is belong to prime/integer power of the prime number/even number
2. Select q which possess 2^q such that $n^2 \leq q < 2n^2$
3. Randomly select integer x , coprime of n , if $\text{coprime}(n, x) = \text{true}$, then $\text{GCD}(n, x) = 1$
4. Perform qubits register allocation as state in fig. 3

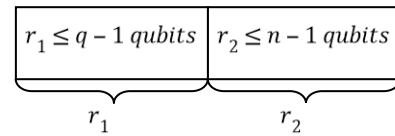


Fig.3. Register r_1, r_2 allocation of $q-1$ and $n-1$ qubits respectively

Hence, $s \leq q - 1 \leq n - 1 \leq t$, where s and t are lower and upper bound of the qubits respectively. An allocated number n is equivalent to $q-1$ and $n-1$ for s and t respectively.

b. Classical Computation Parts of Shor's Algorithm

5. Filled registers r_1 and r_2 states as indicated in fig. 4.

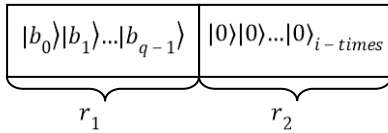


Fig.4. Allocation of qubits registers r_1 and r_2 with $|b_{q-1}\rangle$ and $|0\rangle_{i\text{-times}}$ vector

Where $|b_1\rangle|b_2\rangle\dots|b_{q-1}\rangle$ represent superposition of integer from 0 to $q-1$

The total state of the quantum memory registers at this point is

$$\frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} |b,0\rangle \tag{12}$$

where $|b,0\rangle = \{0, |b_1,0\rangle|b_2,0\rangle\dots|b_{q-1},0\rangle, q-1\}$

6. The computed state of r_1 cloned into r_2 state as indicated in fig. 5. At the time computation of r_1 , its resultant simultaneously cloned into r_2 (fig.5). The r_1 find the period r , denoted here by state $|c_0\rangle|c_1\rangle\dots|c_{q-1}\rangle$ which cloned into r_2 . Hence $r_1 = r_2$ at this stage.

$ b_0\rangle = x^{b_0 \text{ mod } n}\rangle = c_0\rangle$	$ c_0\rangle$
$ b_1\rangle = x^{b_1 \text{ mod } n}\rangle = c_1\rangle$	$ c_1\rangle$
\vdots	\vdots
$ b_{q-1}\rangle = x^{b_{q-1} \text{ mod } n}\rangle = c_{q-1}\rangle$	$ c_{q-1}\rangle$

Fig.5. Computation of $|x^{b_{q-1} \text{ mod } n}\rangle$ and allocation of it into r_2 at the same time

Perform an operation on r_1 and r_2 at the same time due to the parallelism. Hence, quantum memory registers state as

$$\frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} |b, x^b \text{ mod } n\rangle \tag{13}$$

7. Measures $|c_1\rangle|c_2\rangle\dots|c_{q-1}\rangle$ of register r_2 with some value k , due to measurement effect, register r_1 collapsed in one value due to quantum mechanics principle. Therefore, r_1 as in superposition states corresponding with r_2 in between 0 and $q-1$.

The state of the quantum memory registers after this

$$\frac{1}{\sqrt{\|A\|}} \sum_{a'=a \in A} |a', k\rangle \tag{14}$$

Where A is the set of a 's such that $x^a \text{ mod } n = k$, and $\|A\|$ is the number of elements in that set.

8. Now compute the discrete Fourier transform on r_1 . The discrete Fourier transform when applied to a state $|b\rangle$ changes it in the following manner

$$|b\rangle = \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c\rangle * e^{\frac{2\pi i b c}{q}} \tag{15}$$

This step is performing by the quantum computer in one step by using quantum parallelism. After the discrete Fourier transform register is in the state

$$|b\rangle = \frac{1}{\sqrt{\|A\|}} \sum_{b \in A} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c, k\rangle * e^{\frac{2\pi i b' c}{q}} \tag{16}$$

9. Measure state r_1 as $m = |c_1\rangle|c_2\rangle\dots|c_{q-1}\rangle$ and $m = \frac{q}{r} = \text{rem}(0)$, where r is desired period of Shor's algorithms.

c. Classical Computation Parts of Shor's Algorithm

10. With a value of m , on a classical computer do some post-processing which calculates r based on knowledge of m and q . In particular

- m has a high probability of being $\lambda * (q/r)$ where λ is an integer
- If perform floating point division on m/q then calculate the best rational approximation for m/q whose denominator is less than or equal to q
- Take this denominator to be a candidate for r .
- If candidate r is odd then either double it, if doing so leads to a value less than q

There are efficient ways to do this post-processing on a classical computer.

11. Once attained r , a factor n determined by taking $\text{gcd}(x^{r/2} + 1, n)$ and $\text{gcd}(x^{r/2} - 1, n)$. If found a factor of n , then stop, if not go to step 4. This final step is done on a classical computer.

Step 11 contains a provision for what to do if Shor's algorithm failed to produce factors of n . There are a few reasons to fail Shor's algorithm. For example, the discrete Fourier transform could be measured 0 in step 9, making the post-processing in step 10 impossible. The algorithm will sometimes find factors 1 and n , which is not useful either. For these reasons step 11 must be able to jump back to step 4 to start over.

a. Computation Complexity of Shor's Algorithm and Classical

Briefly, $N=p.q$, $N \notin \{\text{Prime}(N) \text{ or } (\text{prime}(N))^2\}$ and $p, q \in \text{Prime of } (N)$ with some constraints. It is included into NP problem in classical computer and not solvable

by polynomial time $P(t)$. But in a quantum computer, it is solvable in polynomial time $P(t)$. As of 2015 the fastest algorithm publicly available for factoring large number runs in $\frac{6.4}{9} O(\exp(n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}))$, where n is the number of bits used to represents the number. This runtime is not in polynomial time. In contrast, Shor's algorithm runs $O((\log n)^2 \times \log \log n)$ on a quantum computer, took $O(\log n)$ -time as post-processing on a classical computer [12]. Overall, it is polynomial.

b. Classical Algorithms Counterpart

The Shor's algorithm is versatile for factor large integer number due to superposition as well as quantum Fourier transforms features. After analysis, it seems suitable for factor large integer number n into p and q . However, most of the computation performed through universal Turing machine whereas rest of the operation performed by quantum Turing machine. For factorization of large integer number require tremendous computation speed which seems impossible to highly classical computers. By Moore's law, the future high-speed computer also not able to crack factorization cryptography as well as further miniaturization seems impossible. The best alternative is a quantum computer to break RSA cryptography and others security mechanism.

VI. RESULT AND ANALYSIS

From above discussion, it is clear that, quantum computer speed become tremendous and processed huge chunk of information within a fraction of time. Deutsch-Jozsa's to Shor's algorithms tells the story of quantum computer processing speed. When consider classical computer then Moore's law notably important in terms of its processing speed. According to Moore's law, classical computer processing speed threshold at specific points due to enforcement of classical physics a constraint which is shown in fig. 6. On such situation quantum computer seems appropriate alternative due to application of quantum mechanics features which vividly shown in fig. 7. Classical processing speed follows exponential whereas quantum processing speed follows straight line. Due to this reason well-known Shor's quantum algorithm computation performs simultaneously by classical as well as quantum computer. Most of the computation carried out by classical computer such as finding period r done in classical computer, rest of the computation performs by quantum computation. Hence, by this Shor's compute in polynomial time (fig.9).

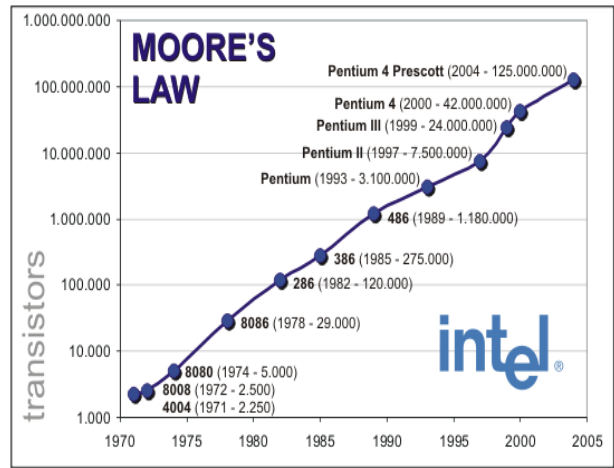


Fig.6. "Moore's Law has been valid since it was first formulated in 1965" (Courtesy: http://www.phonearena.com/news/Moores-Law-is-coming-to-an-end_id54127)

Similarly, while we consider the Shor's quantum algorithm with best classical algorithm then we find that Shor's factorized large integer number into polynomial time whereas best classical algorithm takes exponential time. Shor's factorization takes just a fraction of time, contrary to it best classical algorithm takes hundreds of years to decode factorization terms which is shown in fig. 8.

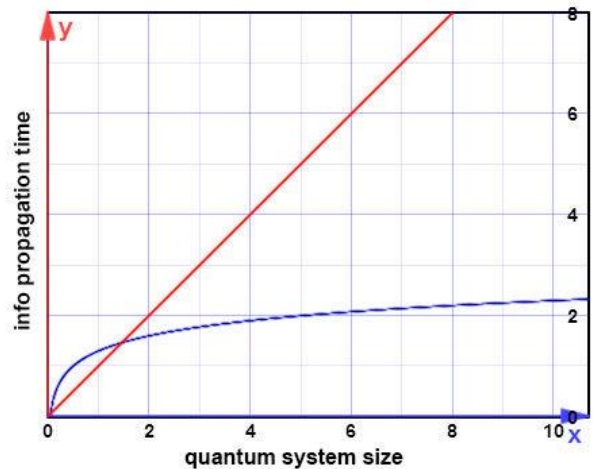


Fig.7. "The size of a quantum computer affects how quickly information can be distributed throughout it. The relation was thought to be logarithmic (blue). Progressively larger systems would need only a little more time. New findings suggest instead a power law relationship (red), meaning that the "speed limit" for quantum information transfer is far slower than previously believed." (Credit: <https://www.nist.gov/news-events/news/2015/04/nist-tightens-bounds-quantum-information-speed-limit>)

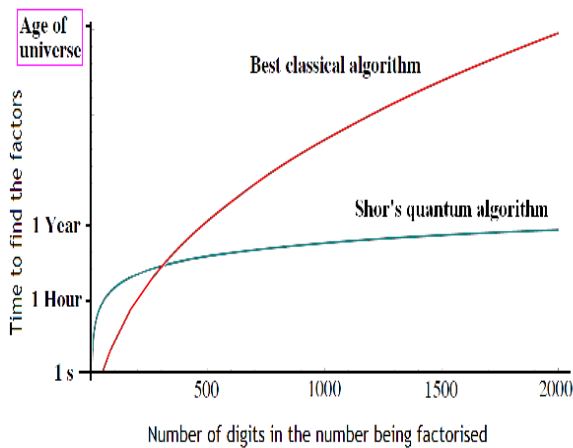


Fig.8. comparative analysis of factorization of integer number by Shor's quantum algorithms and existing best classical algorithm (Courtesy: <http://quantumj13.imascientist.org.uk/profile/chrismanseil/>)

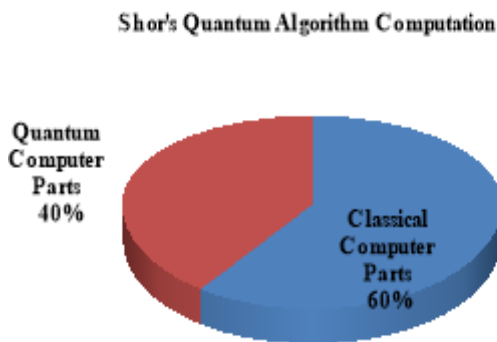


Fig.9. Shor's quantum algorithm computation strategies

VII. CONCLUSION

From above discussion we turn out that existing quantum algorithms processing are in polynomial time. While counterpart classical algorithms not achieved as such performance. Even high processing computing seems impossible to archive as such performance. The classical algorithms performance is in the range of exponential, semi-exponential or quadratic ways for P or NP problems. Therefore, if quantum algorithms like Shor's algorithm work into quantum computer or simulate into deterministic Turing machine as per Church-Turing thesis then existing vulnerabilities into classical cryptography completely remove. Find a new direction for the transactions which gives guarantee about absolute security into classical computer world.

I feel disappoint to mention here that development pace of quantum algorithms as well as quantum computer has not been in full swing. The important cause of it is digital intelligent quotient hurdles to quantum computer development. However, a few things going to good such as, China has launched first quantum satellite in 2016. Similarly, D-wave quantum has been developed which work as per quantum mechanics laws.

ACKNOWLEDGEMENT

Authors acknowledge DG & DY DG of Raksha Shakti University, Ahmedabad for allowing this research work at the aegis of Institute of R&D, Raksha Shakti University, Ahmedabad & we are grateful for their support and guidance.

REFERENCES

- [1] Nielsen M. A., Chuang I. L.: *Quantum Computation and Quantum Information*. 3rd edn, Cambridge Press, UK, 2000
- [2] David Deutsch: *Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer*. Proceedings of the Royal Society of London A 400: 97, 1985
- [3] David Deutsch, Richard Jozsa: *Rapid solutions of problems by quantum computation*. Proceedings of the Royal Society of London A 439: 553, 1992
- [4] Simon, D.R.: *On the power of quantum computation*. Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium, pp. 116–123, 1994
- [5] Lov K. Grover: *A fast quantum mechanical algorithm for database search*. Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212–219, 1996. *arXiv:quant-ph/9605043*
- [6] Peter W. Shor: *Algorithms for quantum computation: discrete logarithms and factoring*. Proceedings of the 35th IEEE Symposium on Foundations of Computer Science. pp. 124–134, 1994
- [7] E.Horowitz, Sahni, D.Mehta: *Fundamentals of Data Structures in C++*. Galgotia Publications
- [8] E. Bernstein, U. Vazirani: *Quantum complexity theory*. Proceeding of the 25th Annual ACM Symposium on Theory of Computing, pp. 11-20, 1993
- [9] E. Bernstein, U. Vazirani: *Quantum complexity theory*. SIAM Journal on computing, 26(5), pp. 1411-1473, 1997
- [10] Scott Aaronson: *Quantum Complexity Theory*; Lecture note 4.
- [11] Pulak Ranjan Giri, Vladimir E. Korepin: *A Review on Quantum Search Algorithms*. pp. 1-43, 2016. <http://arxiv.org/abs/1602.02730v>
- [12] Lomonaco, Jr: *Shor's Quantum Factoring Algorithm*. This paper is a written version of a one-hour lecture given on Peter Shor's quantum factoring algorithm. 22 pages, 2000 *arXiv:quant-ph/0010034*
- [13] C.P. Williams, S.H. Clearwater: *Explorations in Quantum Computing*. 1998
- [14] Arish Pitchai, A V Reddy, Nickolas Savarimuthu, *Quantum Walk Algorithm to Compute Subgame Perfect Equilibrium in Finite Two-player Sequential Games*, International Journal of Mathematical Sciences and Computing(IJMISC), Vol.2, No.3, pp.32-40, 2016.DOI: 10.5815/ijmsc.2016.03.03
- [15] G. Brassard, N. Lütkenhaus, T. Mor, B. C. Sanders: *Limitations on practical quantum cryptography*. Physical Review Letters, 85(6):1330+, 2000
- [16] Einstein A., B. Podolsky, N. Rosen: *Can Quantum-Mechanical Description of Physical Reality be Considered Complete?* Physical Review 47 (10), pp. 777–780, 1935

Authors' Profiles



Shyam R. Sihare is the Ph. D. candidate in Raksha Shakti University, Ahmedabad, India. He took up his Master's degree in Computer Science at Nagpur University, Nagpur, India in 2003 and obtained M. Phil. in Computer Science at Madurai Kamraj University, Madurai, India. He cleared Professor Eligibility Test GSLET, Gujarat,

India in 2011. He obtained MCA at IGNOU, New Delhi, India in 2011.

He is currently working as Asstt. Professor in Computer Science and Application in Dr. APJ Abdul Kalam Govt. College, Silvassa, Dadra & Nagar Haveli(UT), India. His research interests include Quantum Computer, Quantum Algorithms, Quantum Cryptography, and Classical Computer Algorithms.



Dr. V V Nath, earned his Doctor of Philosophy at North Gujarat University, Patan, Gujarat, India in 2012.

He currently works as a professor at the Nirma Institute of Management, Nirma University, Ahmedabad, India. He has 27 years experience in Industry before joining the Institute in June, 2003. His Industry experience spans across various Industry segments like Electronic, Pharmaceutical, Glass Manufacturing, Steel Manufacturing etc. He has been heading Information Technology Department at various Companies in his career with Industry. His Research interest includes Enterprise Systems and Information Security.

How to cite this paper: Shyam R. Sihare, V V Nath, "Analysis of Quantum Algorithms with Classical Systems Counterpart", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.9, No.2, pp.20-26, 2017. DOI: 10.5815/ijieeb.2017.02.03