Primality Testing in Quantum Domain

Kaushik Chakraborty[1] * Anupam Chattopadhyay[2] † Pratyay Poddar[3] ‡

[1] *Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India*
[2] *UMIC Research Centre, RWTH Aachen University, 52074 Aachen, Germany.*
[3] *Hitachi Cambridge Laboratory, Cambridge CB3 0HE, Great Britain.*

Abstract. This paper discusses and shows improvements of Quantum algorithms' efficiency for primality testing algorithm. In the domain of existing classical primality testing algorithm, we point out where massive computation power of quantum Turing machines can be directed for a significant increase in computational speed.

Keywords: Primality testing, Amplitude amplification, AKS algorithm.

1 Introduction & Motivation

In terms of efficiency quantum algorithms outperform best known classical algorithms in several scenarios. Performance of quantum algorithms has been studied in many domains of both computer science and mathematics. In recent years, researchers have started to solve the mysteries of prime numbers using the computation power of quantum Turing machine [9]. This paper is another approach in that direction.

Prime numbers always baffled mathematicians around the globe. In modern times, they play a key role in classical cryptography. Public key cryptography (RSA) is a highly useful technique in classical cryptography, and it requires large prime numbers [2]. However it is extremely difficult to say if a large number is prime or not. The process of finding out whether a given number is prime or not is known as primality testing (PT). There are several randomized polynomial time algorithm techniques available for PT viz. *Miller-Rabin Test*, *Solovay-Strassen Test* [2], [4] etc.

Until 2002, existence of a deterministic polynomial time algorithm for primality testing was not confirmed. In that year, Agarwal *et al.* came up with an affirmative answer to that [1]. This algorithm is known as AKS primality testing algorithm after its inventors' names.

AKS algorithm is based on Fermat's little theorem [2]. The main step of AKS algorithm [1] is following,

Theorem 1 *Given an integer $n > 1$, let r be an integer such that $o_r(n) > \log^2 n$. Suppose n satisfies Equation 1, n has a prime factor $\leq r$ or n is a prime number.*

$$(x+a)^n \equiv x^n + a \ (mod\ x^r - 1, n) \quad for\ a = 1, ..., \lfloor \sqrt{\phi(r)} \log n \rfloor. \tag{1}$$

Running time of AKS algorithm is $O(r^{1.5} \log^3 n)$, where r is a parameter of the algorithm. Agarwal *et al.* showed that the value of r is of $O(\log^5 n)$ [1]. Therefore running time of that algorithm is $O(\log^{10.5} n)$. Many improvements were proposed on the value of r later on [3], [5]. In every improvement of AKS algorithm, there is a loop for testing some condition λ which is satisfied for each $1 \leq a \leq g(r)$, where g is a real valued function.

This paper, assuming the availability of the oracle U for performing the conditioning operation λ, points out those iterations and used amplitude amplification algorithm. Use of amplitude amplification algorithm reduces the number of times the oracle operator needs to be called or the condition checking operator for the operation λ quadratic times.

This article uses original AKS algorithm as an example and shows how we get quadratic improvements from classical AKS algorithm.

2 Quantum Algorithm for Primality Testing

As described in section 1, proposed algorithm is based on classical AKS algorithm. Here, we have used quantum amplitude amplification technique as a subroutine for the main computational step of AKS algorithm. Structure of AKS algorithm allows us to apply quantum amplitude amplification algorithm directly. The main computational task of AKS algorithm is stated in Theorem 1, where the parameter $o_r(n)$ is the smallest number j for which,

$$n^j \equiv 1 \ (mod\ r) \tag{2}$$

*kaushik.chakraborty9@gmail.com
†Anupam.Chattopadhyay@ice.rwth-aachen.de
‡pp374@cam.ac.uk

$\phi(r)$ is the Euler's totient function of r [2].

To compute equation 1 using quantum amplitude amplification algorithm, we call a subroutine named **QPrime**, which is an equal superposition state of all a's such that $a \in \{0, \lfloor\sqrt{\phi(r)}\log(n)\rfloor\}$. Then it is assumed that there exists an oracle operator U_P, which has the following characteristics:

$$\begin{aligned} U_P|a\rangle &= -|a\rangle \quad if(X+a)^n \not\equiv X^n + a \ (mod\ X^r - 1, n) \quad \& \quad a \neq 0 && (3) \\ &= \quad |a\rangle \quad otherwise && (4) \end{aligned}$$

QPrime will return 0, *iff* $\nexists\ a \in \{1, \lfloor\sqrt{\varphi(r)}\log(n)\rfloor\}$ such that $(X+a)^n \not\equiv X^n + a\ (mod\ X^r - 1, n)$. This subroutine uses quantum amplitude amplification technique for checking the existence of such a(s). Using generalized version of quantum amplitude amplification, we solve this problem [7], [8].

QPrime subroutine is a typical amplitude amplification algorithm setting, and our goal here is to test whether the function P corresponding U_P is a constant or not. This results in quadratic speed up compared to the classical sequential searching. So, in a quantum setting it will take $O(\lceil(\phi(r))^{\frac{1}{4}}(\log n)^{\frac{1}{2}}\rceil)$ evaluations of U_P instead of $O(\lfloor\sqrt{\phi(r)}\log n\rfloor)$, which is a quadratic speed up.

One can use BBHT [7] algorithm instead of **QPrime** to find the existence of such a(s).

3 Conclusion

Quantum reversible circuit realisation of the operation $(X+a)^n \equiv X^n + a\ (mod\ X^r - 1, n)$ with optimal number of elementary gates may also reduce actual time complexity of original AKS algorithm.

Our future work includes studying the realization of U_P operator for AKS algorithm and also circuit realization of the corresponding oracle operators for implementing the condition operation λ.

References

[1] Agarwal Manindra and Kayal Neeraj and Saxena Nitin, PRIMES is in P, 2004 doi=10.4007/annals.2004.160.781

[2] Stinson Douglas, In*Cryptography: Theory and Practice,Second Edition, 2006*, year = 2006, isbn = 1584882069, edition = 3rd, publisher = CRC/C&H

[3] Tsz-wo Sze, A Potentially Fast Primality Test, 2007

[4] Goldwasser Shafi and Kilian Joe, Primality Testing using Elliptic Curves In*J. ACM,* 46(4):450-472,1999

[5] Lenstra, W. Hendrik Jr. and Pomerance Carl, Primality Testing with Gaussian Periods Available at `http://www.math.dartmouth.edu/~carlp/PDF/complexity12.pdf` 2005.

[6] Brassard, Gilles and Høyer, Peter and Mosca, Michele and Tapp, Alain, Quantum amplitude amplification and estimation, In *Quantum computation and information (Washington, DC, 2000)*, volume 305, pages 53–74, Amer. Math. Soc., 2002.

[7] Boyer, Michel and Brassard, Gilles and Høyer, Peter and Tapp, Alain, Tight Bounds on Quantum Serching, In*Fortsch. Phys. 46:493-506,1998* Available at `http://arxiv.org/abs/quant-ph/9605034v1`

[8] Chakraborty, Kaushik and Maitra, Subhamoy Quantum algorithm to check Resiliency of a Boolean function, In*IACR Cryptology ePrint Archive,2013*, volume 2013, pages 232.

[9] Latorre I. Jose and Sierra, German, Available at `http://arxiv.org/abs/1302.6245v2`