3 Shor のアルゴリズムのための効率的な量子回路

高橋 康博〔日本電信電話（株）NTT コミュニケーション科学基礎研究所〕

1994 年，AT&T Bell 研究所の Shor は，因数分解問題や離散対数問題を効率的に解く量子コンピュータ上のアルゴリズムを提案した．これ以来，Shor のアルゴリズムを実行するための効率的な量子回路の理論的研究が行われている．本稿では，量子ビット数の最小化に焦点を当て，このような研究の最新の成果について紹介する．

はじめに

1994 年，AT&T Bell 研究所の Shor（現 MIT）は，因数分解問題や離散対数問題を効率的に解く量子コンピュータ上のアルゴリズムを提案した[7]．現在のコンピュータではこれらの問題を効率的に解くことは困難であると考えられているため，量子コンピュータは現在のコンピュータよりも高い計算能力を持つと考えられる．したがって，量子コンピュータは有用なコンピュータになり得ると予想される．一方で，量子コンピュータにより，RSA 暗号など一般に使われている暗号のいくつかは破られてしまう．これは，このような暗号の安全性が因数分解問題や離散対数問題の困難さに根拠を置いているからである．

現在，量子コンピュータの実現に向けて多くの研究が行われている．しかし，現状では量子コンピュータが暗号にとって脅威となっているわけではない．これは，多くの計算資源（量子ビット数や計算時間等）を持つ量子コンピュータが実現されていないからである．それでは，どの程度の計算資源を持つ量子コンピュータが実現すると暗号にとって脅威となるのであろうか．言い換えると，どの程度少ない計算資源で Shor のアルゴリズムが実行できるのであろうか．このような問題を扱うのが Shor のアルゴリズムのための効率的な量子回路の研究である．量子コンピュータの実現に向けた研究が進むにつれて，量子コンピュータに対する現在の暗号の安全性を詳細に分析し，量子コンピュータに対しても安全な暗号を構築することが大きな課題となってくる．Shor のアルゴリズムのための効率的な量子回路の研究は，このような課題の解決に多くの知見を与える重要な研究である．

Shor のアルゴリズムが提案されて以来，焦点が当てられてきた問題は，どの程度少ない量子ビットで因数分解問題を解くことができるのかという問題である．詳細は次の章で述べるが，この問題は，どの程度少ない量子ビットで量子フーリエ変換の逆演算とべき乗剰余演算が実行できるのかという問題に帰着される．量子フーリエ変換の逆演算を実行するためには 1 量子ビットあれば十分であることが知られているため[5]，以下の議論で中心となるのはべき乗剰余演算である．n ビットで表される数 N を因数分解したい場合，Shor のアルゴリズムにおいて使われるべき乗剰余演算は，

$$|x\rangle|1\rangle \rightarrow |x\rangle|a^x \bmod N\rangle$$

という演算であり，x は n ビットで表される数，a は $1<a<N$ を満たす事前に与えられた数である．

1996 年，Vedral らは，少ない量子ビットでべき乗剰余演算を行う方法を提案し，n ビットで表される数の因数分解が $7n+1$ 個の量子ビットで行えることを示した[10]．その後，上で触れた量子フーリエ変換の逆演算を行う方法[5]や少ない量子ビットで加算を行う方法[2]が提案された．これらの方法と Vedral らのべき乗剰余演算を行う方法を組み合わせ，さらに，事前に与えられた数を量子回路に“組み込む”ことにより，2003 年に Beauregard が $2n+3$ 個の量子ビットで因数分解が行えることを示した[1]．2006 年には Takahashi らが，Beauregard の量子回路における未使用な量子ビットを利用し，$2n+2$ 個の量子ビットで因数分解が行えることを示した[9]．これは，

現在推奨されている 1024 ビットの鍵の長さを持つ RSA 暗号が，2050 個の量子ビットを持つ量子コンピュータで理論上破られることを示している．

本稿では，Beauregard の量子回路とそれに関連する最新の成果について紹介する．以下では，まず始めに Shor のアルゴリズムのための量子回路と 1 量子ビットだけを使って量子フーリエ変換の逆演算を行う方法について述べる．次に，Beauregard の量子回路について，Vedral らのべき乗剰余演算を行う方法も含めて詳細に述べる．次に，関連する話題として，Takahashi らによる Beauregard の量子回路の改良と Fowler らの隣接量子ビット間の演算だけを使う Shor のアルゴリズムのための量子回路[3]について触れる．隣接量子ビット間の演算だけを使う量子回路は，量子コンピュータを実現する際の現実的な制限を考慮した量子回路である．最後に，今後の課題について述べる．

Shor のアルゴリズムのための量子回路

❒ Shor のアルゴリズム

因数分解問題は，n ビットで表される合成数 N に対し，N の自明でない（すなわち，1 と N 以外の）約数を求める問題である．Shor のアルゴリズムはこの問題を効率的に解くが，このアルゴリズムは，現在のコンピュータによって実行される部分と量子コンピュータによって実行される部分に分けられる．量子コンピュータによって実行されるのは，位数発見問題を解く部分である．位数発見問題は，因数分解したい合成数 N と N より小さく N と互いに素な自然数 a に対し，N を法とする a の位数を求める問題である．ただし，N を法とする a の位数は，

$$a^r = 1 \bmod N$$

となる最小の自然数 $r > 0$ である．

位数発見問題を解くアルゴリズムは次のようなものである．$2n$ 個の量子ビットからなる第 1 レジスタと n 個の量子ビットからなる第 2 レジスタを用意する．初期状態として，第 1 レジスタが 0 を表し，第 2 レジスタが 1 を表しているとする．

(1) 第 1 レジスタの各量子ビットにアダマール変換 H を適用し，第 1 レジスタに 0 から $2^{2n}-1$ までの 2^{2n} 個の値の重ね合わせを作る．

(2) 第 1 レジスタに重ね合わせられた各値 x に対するべき乗剰余 $a^x \bmod N$ を計算し，第 2 レジスタに値を書き込む．

(3) 第 1 レジスタに量子フーリエ変換 QFT の逆演算 QFT^{-1} を適用する．

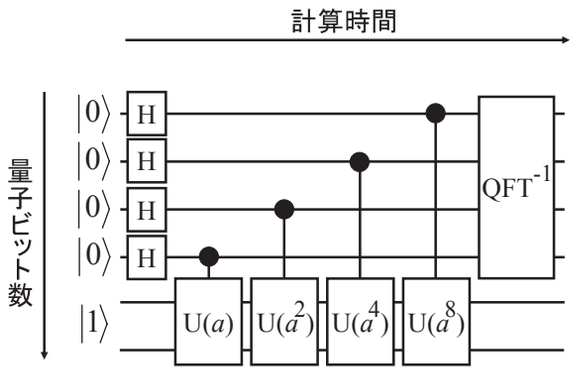

図 -1　Shor のアルゴリズムのための量子回路

(4) 第 1 レジスタを観測する．

第 4 ステップで得られた値を使い，現在のコンピュータにより効率的に位数を求めることができる．

第 2 ステップにおけるべき乗剰余は，第 1 レジスタの各量子ビットの値に依存して，剰余乗算 $\mathrm{U}(a^{2^i})$ $(i = 0,\ldots, 2n-1)$ を第 2 レジスタに適用することにより計算される．ただし，$\mathrm{U}(a)$ は，

$$\mathrm{U}(a)|x\rangle = |ax \bmod N\rangle$$

という演算であり，x は n ビットで表される数である．a と N は事前に与えられていることに注意する．

❒ 量子回路

量子回路は，量子コンピュータ上の各量子ビットにどのような演算を適用して計算を進めるかを表したものである．量子回路の効率性は，入力の長さに対し，その量子回路で使われる量子ビット数，基本演算数，そして計算ステップ数がどのように表されるかによって測る．基本演算は 1 量子ビットに対する演算と 2 量子ビットに対する演算であり，基本演算数はその量子回路で使われている基本演算の数である．計算ステップ数は，その量子回路で使われている基本演算を並列に行える演算からなるグループに分割したときのグループの数である．ただし，異なる量子ビットに対する演算は並列に行えると仮定する．大まかには，基本演算数は計算時間に対応し，計算ステップ数は各基本演算を可能な限り並列に処理した場合の計算時間に対応する．

Shor のアルゴリズムのための量子回路は，上で述べた位数発見問題を解くアルゴリズムのための量子回路である．入力の長さは，因数分解したい合成数を 2 進数で表したときの長さ n である．この量子回路は図 -1 のようになる．ただし，$n = 2$ の場合である．1 本の横の線は 1 個の量子ビットに対応しており，左から右へ向けて計算が進む．第 1 レジスタは上から $2n$ 個の量子ビ

ットからなり，第 2 レジスタはその下 n 個の量子ビットからなる．初期状態は最も左に記述している．図 -1 では，始めに第 1 レジスタの各量子ビットに H が適用されているが，これはアルゴリズムのステップ 1 に対応する．次に適用されている黒丸とそれにつながった U(a) は，黒丸が置かれている量子ビットの値に依存して第 2 レジスタに U(a) が適用されていることを表しており，その後の演算は U(a^2) 等が同様に適用されていることを表している．これはアルゴリズムのステップ 2 に対応する．最後に第 1 レジスタに QFT^{-1} が適用されているが，これはアルゴリズムのステップ 3 に対応する．アルゴリズムでは最後に第 1 レジスタを観測するが，量子回路の効率性とはかかわらないので図 -1 では省略している．

図 -1 の量子回路の構成から，Shor のアルゴリズムのための量子回路の効率性は，U(a) と QFT^{-1} のための量子回路の効率性により決まることが分かる．したがって，Shor のアルゴリズムのための効率的な量子回路を構成するためには，U(a) と QFT^{-1} のための効率的な量子回路を（基本演算を使って）構成すればよい．以下では，少ない量子ビットで Shor のアルゴリズムのための量子回路を構成するという問題を扱うが，これは，少ない量子ビットで U(a) と QFT^{-1} のための量子回路を構成するという問題に帰着される．

❒ 少ない量子ビットで QFT^{-1} を行う方法

Mosca らは，Shor のアルゴリズムのための量子回路において，少ない量子ビットで QFT^{-1} を行う方法を提案した [5]．この方法は，観測を交えることにより，**図 -2** のように 1 量子ビット上で QFT^{-1} を適用する．したがって，第 1 レジスタは 1 量子ビットあれば十分であるということになる．

図 -1 と図 -2 における演算の対応関係は次のようになる．図 -2 では，始めに第 1 レジスタに H が適用されているが，これは図 -1 の最も上の量子ビットに適用されている H に対応する．次に第 1 レジスタの量子ビットの値に依存して第 2 レジスタに U(a^8) が適用されているが，これは図 -1 の第 1 レジスタの量子ビットの値に依存して第 2 レジスタに適用されている U(a^8) に対応する．次に第 1 レジスタに H が適用されているが，これは図 -1 の QFT^{-1} の中で最も上の量子ビットに適用されている H（図 -1 では簡単のために省略している）に対応する．次の第 1 レジスタの観測は，図 -1 の QFT^{-1} の後に最も上の量子ビットを観測することに対応する．次に第 1 レジスタに 1 量子ビットに対する演算 U_1 が適用されているが，これは先ほどの観測結果に応じて第 1 レジスタを 0 に戻すための演算であり，これに対応する図 -1 の演算はない．このような操作を繰り返すことにより図 -1 と同等の量子回路が構成できる．したがって，少ない量子ビットで Shor のアルゴリズムのための量子回路を構成するためには，少ない量子ビットで U(a) のための量子回路を構成すればよい．

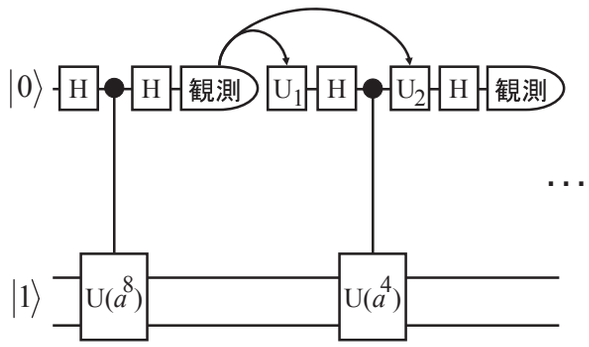

図 -2　少ない量子ビットで QFT^{-1} を行う方法

Beauregard の量子回路

❒ U(a) の分解

Beauregard は，少ない量子ビットで U(a) のための量子回路を構成するために，Vedral らの方法 [10] に従って U(a) を単純な演算に分解した．この方法では，U(a) を剰余乗加算 V(a) に分解し，V(a) を剰余加算 W(a) に分解する．ただし，V(a) は，

$$\mathrm{V}(a)|x\rangle|b\rangle = |x\rangle|ax + b \bmod N\rangle$$

という演算であり，x と b は n ビットで表される数である．また，W(a) は，

$$W(a)|c\rangle|b\rangle = \begin{cases} |c\rangle|a+b \bmod N\rangle & c=1\text{のとき} \\ |c\rangle|b\rangle & c=0\text{のとき} \end{cases}$$

という演算であり，c は 0 または 1，b は n ビットで表される数である．

より正確には，U(a) は V(a) と V(a^{-1}) とスワップ演算を使って次のように計算できる．

$$\begin{aligned} |x\rangle|0\rangle &\rightarrow |x\rangle|ax \bmod N\rangle \\ &\rightarrow |ax \bmod N\rangle|x\rangle \\ &\rightarrow |ax \bmod N\rangle|x - a^{-1}ax \bmod N\rangle \\ &= |ax \bmod N\rangle|0\rangle \end{aligned}$$

最初の演算が V(a) であり，2 番目の演算はスワップ演算であり，3 番目は $\mathrm{V}(a^{-1})^{-1}$ である．ただし，a^{-1} は N を法とする a の逆元である．a と N が互いに素であることから a^{-1} は存在し，ユークリッドのアルゴリズムにより現在のコンピュータを使って効率的に a^{-1} を求めることができる．また，V(a) は，

$$ax + b \bmod N$$
$$= (2^{n-1}ax_{n-1} + (\cdots (2^1 ax_1 +$$
$$(2^0 ax_0 + b \bmod N) \bmod N)\cdots) \bmod N)$$

という関係が成り立つので，$W(2^i a)$ $(i = 0,\ldots, n-1)$ を使って計算できる．ただし，$x_{n-1} \ldots x_0$ は x を 2 進数で表したものである．したがって，少ない量子ビットで $U(a)$ のための量子回路を構成するためには，少ない量子ビットで $W(a)$ のための量子回路を構成すればよい．

❐ W(a) のための量子回路

Beauregard の $W(a)$ のための量子回路は次のようなアルゴリズムに基づいている．ただし，$c = 1$ と仮定し，c を含むレジスタは省略する．初期状態は $|b\rangle$ である．

(1) a を b に加える．結果は $|b + a\rangle$ となる．
(2) $b + a$ から N を引く．結果は $|b + a - N\rangle$ となる．
(3) $b + a - N < 0$ であるかどうかを判定するため，$b + a - N$ の最上位ビット y を 1 つの補助ビットに書き込む．結果は $|b + a - N\rangle|y\rangle$ となる．ただし，$b + a - N < 0$ ならば y は 1 であり，そうでなければ 0 である．
(4) y が 1 ならば N を $b + a - N$ に加える．結果は $|a + b \bmod N\rangle|y\rangle$ となる．
(5) $a + b \bmod N$ から a を引く．結果は $|(a + b \bmod N) - a\rangle|y\rangle$ となる．
(6) $(a + b \bmod N) - a < 0$ であるかどうかを判定するため，$(a + b \bmod N) - a$ の最上位ビット z の否定を補助ビットに書き込む．結果は $|(a + b \bmod N) - a\rangle|y \oplus z \oplus 1\rangle$ となる．ただし，$(a + b \bmod N) - a < 0$ ならば z は 1 であり，そうでなければ 0 である．$\oplus$ は 2 を法とする加算を表す．
(7) a を $(a + b \bmod N) - a$ に加える．結果は $|a + b \bmod N\rangle|0\rangle$ となる．

ステップ 6 において，

$$y \oplus z \oplus 1 = 0$$

であることに注意する．なぜなら，

$$a + b \bmod N \geq a \Leftrightarrow a + b < N$$

という関係が成り立つからである．Beauregard の $W(a)$ のための量子回路は，上のアルゴリズムにおける加算や減算を以下で述べる QFT を使う加算のための量子回路により実現したものである．

❐ QFT を使う加算のための量子回路

Draper は，加算 ADD のための効率的な量子回路を提案した [2]．ただし，ADD は，

$$\mathrm{ADD}|x\rangle|y\rangle|0\rangle = |x\rangle|x + y\rangle$$

という演算であり，x と y は n ビットで表される数である．$|0\rangle$ は $x + y$ の最上位ビットのために用意してある．x を含むレジスタを第 1 レジスタと呼び，それ以外の量子ビット全体を第 2 レジスタと呼ぶとき，この量子回路は次のように構成される．

(1) 第 2 レジスタに QFT を適用する．
(2) 第 1 レジスタの各量子ビットの値に依存して，第 2 レジスタの各量子ビットに位相シフト演算を適用する．
(3) 第 2 レジスタに QFT^{-1} を適用する．

この量子回路は，少ない量子ビットで $W(a)$ のための量子回路を構成するために有用な 2 つの性質を持つ．1 つは，第 1 レジスタと第 2 レジスタ以外に量子ビットを必要としないことである．もう 1 つは，x が事前に与えられた（古典的な）数の場合，x を量子回路に“組み込む”ことができることである．すなわち，第 1 レジスタは用意する必要がない．これは，x が第 2 レジスタに位相シフト演算を適用するかどうかを制御することだけに使われているからである．この 2 つの性質を同時に持つ ADD のための量子回路は現時点ではほかに例がない．

この量子回路は，基本演算数については不利であることに注意する．これは，この量子回路の基本演算数が $O(n^2)$ である一方で，多くの量子ビットを使えば基本演算数が $O(n)$ である ADD のための量子回路を構成できるからである [10]．QFT を使う ADD のための量子回路において，近似 QFT を使い，小さい誤り確率を許して基本演算数を $O(n \log n)$ に削減できることが知られている．計算ステップ数は近似 QFT を使うかどうかにかかわらず $O(n)$ である．

以上のことから，Beauregard の量子回路全体の効率性を測ることができる．量子ビット数については，ADD のために $n + 1$ 個，$W(a)$ のために $n + 3$ 個，$V(a)$ のために $2n + 2$ 個，$U(a)$ のために $2n + 2$ 個の量子ビットを使う．第 1 レジスタの 1 個の量子ビットを合わせて，回路全体が使う量子ビットは $2n + 3$ 個である．基本演算数と計算ステップ数は，それぞれ $O(n^3 \log n)$，$O(n^3)$ となる．

関連する話題

❐ Beauregard の量子回路の改良

Takahashi らは，Beauregard の量子回路における未使用な量子ビットを利用し，$2n + 2$ 個の量子ビットを使う Shor のアルゴリズムのための量子回路を構成した [9]．基本演算数は Beauregard の量子回路のおよそ半分である．Takahashi らの量子回路と Beauregard の量子回路の違い

は，W(a) のための量子回路の構成である．Takahashi らの W(a) のための量子回路は次のようなアルゴリズムに基づいている．ただし，$c = 1$ と仮定し，c を含むレジスタは省略する．初期状態は $|b\rangle$ である．

(1) b と $N - a$ の大小を比較し，その結果 y を 1 つの補助ビットに書き込む．結果は $|b\rangle|y\rangle$ となる．ただし，$b < N - a$ ならば y は 1 であり，そうでなければ 0 である．

(2) y が 1 ならば a を b に加え，y が 0 ならば b から $N - a$ を引く．結果は $|a + b \bmod N\rangle|y\rangle$ となる．

(3) $a + b \bmod N$ と a の大小を比較し，その結果 z の否定を補助ビットに書き込む．結果は $|a + b \bmod N\rangle|y \oplus z \oplus 1\rangle = |a + b \bmod N\rangle|0\rangle$ となる．ただし，$a + b \bmod N < a$ ならば y は 1 であり，そうでなければ 0 である．

$y \oplus z \oplus 1 = 0$ となるのは，Beauregard のアルゴリズムの場合と同様である．

計算結果 $a + b \bmod N$ を含むレジスタに必要となる量子ビット数が，Beauregard のアルゴリズムと Takahashi らのアルゴリズムでは異なる．Beauregard のアルゴリズムは減算を使って数の大小比較を行うため，$b + a - N$ のような $n + 1$ ビットで表される可能性のある値を扱う．したがって，そのレジスタに $n + 1$ 個の量子ビットを必要とする．一方，Takahashi らのアルゴリズムはそのような値を扱わないため，n 個の量子ビットで十分である．また，Takahashi らのアルゴリズムで使う加算や減算の回数は，Beauregard のアルゴリズムで使う加算や減算の回数よりも少ないため，基本演算数が削減される．

Takahashi らのアルゴリズムにおける加算，減算，そして大小比較を新たな量子ビットを使わない量子回路で実現すれば，上で述べたことから Takahashi らの量子回路は Beauregard の量子回路と比較して，1 個の量子ビットを削減することができる．加算と減算については QFT を使う量子回路を使えばよい．問題は大小比較のための量子回路を構成することである．より正確には，新たな量子ビットを使わずに COMP(a) のための量子回路を構成することである．ただし，COMP(a) は，

$$\mathrm{COMP}(a)|b\rangle|z\rangle = |b\rangle|z \oplus y\rangle$$

という演算であり，b は n ビットで表される数であり，z は 0 または 1 である．また，$a > b$ であれば y は 1 であり，そうでなければ 0 である．

Takahashi らは Vedral らの ADD のための量子回路 [10] を基に，$|0\rangle$ に初期化されていない $n-1$ 個の補助ビットを使い，COMP(a) のための量子回路を構成した．Beauregard の量子回路における未使用な量子ビットが，このような $n-1$ 個の未初期化補助ビットとして利用できる．より正確には，Beauregard の量子回路において，W($2^i a$) ($i = 0,..., n-1$) を適用している際の x_j ($j \neq i$) を表している量子ビットが利用できる．したがって，新たな量子ビットを使わずに，COMP(a) のための量子回路が構成できる．ページ数の都合によりこの量子回路の詳細は述べられないが，基本演算数と計算ステップ数は共に $O(n)$ である．

❐ 隣接量子ビット間の演算だけを使う量子回路

量子コンピュータの実現に向けた研究において，量子ビットを 1 列に並べ，隣同士の量子ビット間の演算だけを使って計算を進める量子コンピュータの構造が提案されている．これは，1 量子ビットに対する演算と（任意の 2 量子ビット間ではなく）隣接量子ビット間の演算だけを量子回路の基本演算とすることに対応する．このような現実的な計算資源だけを使う量子回路において，どの程度少ない計算資源で Shor のアルゴリズムが実行できるのかを明らかにしようという研究が行われている．

Fowler らは Beauregard の量子回路を基に，隣接量子ビット間の演算だけを使う Shor のアルゴリズムのための量子回路を構成した [3]．基本演算に制限が加えられているため，多くの計算資源を必要とするのではないかと予想されるが，実際には Fowler らの量子回路の効率性は，Beauregard の量子回路の効率性とほぼ同等である．また，上で述べた Takahashi らの改良が隣接量子ビット間の演算だけを使う量子回路においても可能であることを示すことができる．

今後の課題

今後の課題の 1 つは，より現実的な制限のもとで，どの程度少ない計算資源で Shor のアルゴリズムが実行できるのかを明らかにすることである．このような研究については上で触れたが，たとえば，隣接量子ビット間の演算についても現実的に容易に適用できる演算とそうでない演算があるであろう．容易に適用できる演算だけで Beauregard の量子回路と同等の効率性の量子回路が構成できるであろうか．

もう 1 つの課題は，暗号とのかかわりを詳細に分析することである．このような研究の例として Proos らの研究が挙げられる [6]．Proos らは，楕円離散対数問題を解く Shor のアルゴリズムのための効率的な量子回路を構成し，量子コンピュータに対する RSA 暗号の安全性と楕円曲線暗号の安全性について比較した．このような研究においては，本稿で焦点を当てた量子ビット数だけではなく，Kunihiro が行っているような計算時間について

の詳細な分析[4]も重要となる．

本稿では，どの程度少ない量子ビットで因数分解問題を解く Shor のアルゴリズムが実行できるのかという問題に焦点を当て，最新の成果を紹介した．上で述べた以外の関連する研究として，加算のための効率的な量子回路の研究が挙げられる[8]．Los Alamos National Laboratory の量子物理学アーカイブにおいて，関連する多くの興味深い論文を見つけることができる．

謝辞　本稿を作成するにあたり，電気通信大学の國廣昇先生に多くの貴重な助言をいただきました．ここに感謝いたします．

参考文献

1）Beauregard, S. : Circuit for Shor's Algorithm Using 2*n*+3 Qubits, Quantum Information and Computation, Vol.3, No.2, pp.175-185 (2003).
2）Draper, T. G. : Addition on a Quantum Computer, quant-ph/0008033 (2000).
3）Fowler, A. G., Devitt, S. J. and Hollenberg, L. C. L. : Implementation of Shor's Algorithm on a Linear Nearest Neighbour Qubit Array, Quantum Information and Computation, Vol.4, No.4, pp.237-251 (2004).
4）Kunihiro, N. : Exact Analyses of Computational Time for Factoring in Quantum Computers, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E88-A, No.1, pp.105-111 (2005).
5）Mosca, M. and Ekert, A. : The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Computer, Lecture Notes in Computer Science, Vol.1509, pp.174-188 (1999).
6）Proos, J. and Zalka, C. : Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves, Quantum Information and Computation, Vol.3, No.4, pp.317-344 (2003).
7）Shor, P. W. : Algorithms for Quantum Computation : Discrete Logarithms and Factoring, Proc. 35th Annual IEEE Symposium on Foundations of Computer Science, pp.124-134 (1994).
8）Takahashi, Y. and Kunihiro, N. : A Linear-size Quantum Circuit for Addition with No Ancillary Qubits, Quantum Information and Computation, Vol.5, No.6, pp.440-448 (2005).
9）Takahashi, Y. and Kunihiro, N. : A Quantum Circuit for Shor's Factoring Algorithm Using 2*n*+2 Qubits, Quantum Information and Computation, Vol.6, No.2, pp.184-192 (2006).
10）Vedral, V., Barenco, A. and Ekert, A. : Quantum Networks for Elementary Arithmetic Operations, Physical Review A, Vol.54, No.1, pp.147-153 (1996).

（平成 18 年 10 月 24 日受付）

高橋　康博（正会員）
takahasi@theory.brl.ntt.co.jp
2000 年東北大学大学院理学研究科数学専攻修士課程修了．同年 NTT コミュニケーション科学基礎研究所入社．量子計算，計算量理論，暗号理論に興味を持つ．