

ANÁLISIS DEL ALGORITMO CUÁNTICO DE FACTORIZACIÓN DE SHOR

C. L. Mayda¹, J. A. C. Nogales², G. M. Ramírez³

¹Carrera de Informática Universidad Mayor de San Andrés

²³Instituto de Investigaciones Físicas Universidad Mayor de San Andrés. Casilla 8635, La Paz-Bolivia

RESUMEN

Se presenta un análisis del algoritmo de factorización cuántica de Shor en base a pseudo-simulaciones y se explica la comparación con el correspondiente algoritmo clásico.

1. INTRODUCCIÓN

Independientemente del tipo de datos que utilice la computadora y de lo complejos que puedan parecer, estos datos solo existen como unos o ceros a los cuales se denominan *bits* (dígito binario). El bit es la unidad mínima de información que puede ser representado por uno de dos diferentes estados, los cuales pueden ser no o si, falso o verdadero, o simplemente 0 ó 1.

El transistor ha podido representar de manera eficiente (ahorro de energía y fiabilidad) los valores del bit. Sin embargo, el nivel de miniaturización de éste componente está llegando a su límite, donde el umbral cuántico se hace presente y los componentes de las computadoras tendrán que reconocer leyes que escapan a la intuición clásica y en las cuales se introduce el *qubit* o bit cuántico.

Análogamente a la computación clásica, el qubit es la unidad mínima de información y puede ser representado por cualquier sistema cuántico (átomos, fotones, etc.) con dos estados discretos cuya notación es $|0\rangle$ y $|1\rangle$ (notación de brakets⁴). Sin embargo, a diferencia del bit, el qubit puede existir en una superposición de estados, es decir que puede estar en una superposición de 0 y 1 al mismo tiempo. En la Fig. 1 se puede observar la representación gráfica de un qubit con cuatro estados diferentes.

La superposición de estados está representada por:

$$\Psi = a \cdot |0\rangle + b \cdot |1\rangle,$$

donde a y b son números complejos tales que satisfacen la igualdad $|a|^2 + |b|^2 = 1$. Siendo $|a|^2$ la probabilidad de observar el valor $|0\rangle$ y $|b|^2$ la probabilidad de observar el valor $|1\rangle$ [2]. Es decir, que al querer observar la información del qubit, este toma el valor de 0 ó 1 con cierta probabilidad. (Fig. 2)

La superposición de estados puede ser explicado de la siguiente manera:

- (1) Desde un punto de vista físico, el qubit es un vector unitario bidimensional en un espacio vectorial

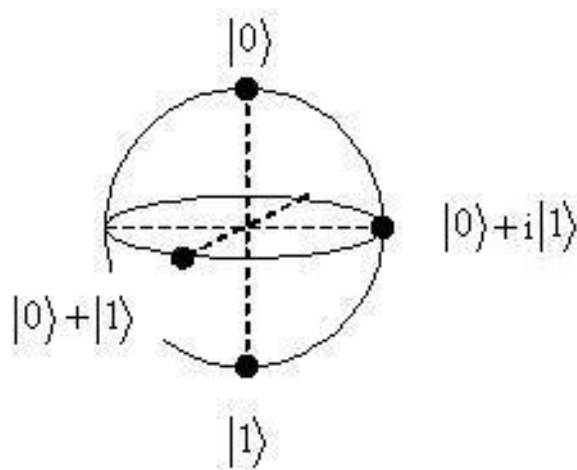


Figura 1. Representación gráfica de cuatro estados de un qubit [1].

complejo, el cual tiene como base $\{|0\rangle, |1\rangle\}$

$$\text{qubit} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- (2) En el álgebra de kets y bras, los *kets* forman un espacio vectorial, donde los elementos del vector son lineales

$$|V\rangle = a_1 |v_1\rangle + a_2 |v_2\rangle + \dots + a_n |v_n\rangle.$$

- (3) Puesto que el qubit es un vector cuya notación está dada por $|\psi\rangle$ (un *ket*), entonces de (1) y (2) se tiene que los elementos del qubit son lineales, es decir:

$$|\text{qubit}\rangle = a |0\rangle + b |1\rangle,$$

obteniendo así el estado superpuesto del qubit.

El artículo explora primeramente algunos conceptos importantes en la Sec. 2; en la Sec. 3, se plantea el problema de factorización y se explica el algoritmo cuántico de Shor con un análisis previo del correspondiente algoritmo clásico. Asimismo, se muestran los resultados obtenidos en términos de tiempos de ejecución tanto del algoritmo clásico como de la pseudo-simulación del algoritmo cuántico. Finalmente, en Sec. 4 se dan las conclusiones y perspectivas del trabajo.

¹e-mail: claudiamayda@gmail.com

²e-mail: jnogales@hotmail.com

³e-mail: gramirez@ulb.ac.be

⁴El álgebra de kets $|\rangle$ y bras $\langle|$ fue introducida por Dirac especialmente para la mecánica cuántica.

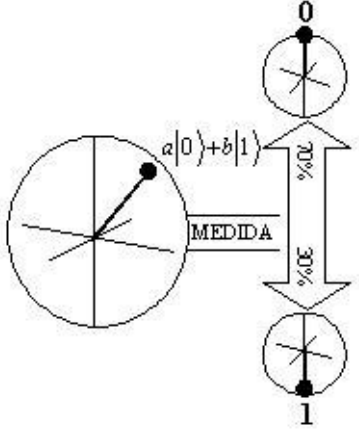


Figura 2. Superposición de estados de un qubit con sus respectivas probabilidades de medida.

2. CONCEPTOS DE COMPUTACIÓN CUÁNTICA

Para poder comprender el análisis del algoritmo cuántico, es necesario tener en claro algunos conceptos que se detallan a continuación.

2.1. Múltiples qubits

Hasta ahora se ha visto la superposición cuántica en un solo qubit, sin embargo, se puede utilizar esta superposición en valores múltiples. A manera de ejemplo, consideremos un registro clásico de tres bits, el cual en un momento dado puede almacenar uno de los ocho valores siguiente:

$$000, 001, 010, 011, 100, 101, 110, 111.$$

A diferencia del registro clásico, un registro de tres qubits puede almacenar en un momento dado y simultáneamente los ocho valores anteriores en una superposición cuántica:

$$\Psi = c_0 |000\rangle + c_1 |001\rangle + c_2 |010\rangle + c_3 |011\rangle + c_4 |100\rangle + c_5 |101\rangle + c_6 |110\rangle + c_7 |111\rangle.$$

Adicionando más qubits al registro, la capacidad de almacenamiento crece exponencialmente, es decir, si tres qubits almacenan 8 valores diferentes, 4 pueden almacenar 16, 5 almacenan 32 y así sucesivamente. En general, L qubits pueden almacenar 2^L números diferentes al mismo tiempo [3]. La mecánica cuántica explica la superposición de estos valores mediante el producto tensorial (\otimes). Sean dos espacios vectoriales V y W bidimensionales con bases $\{v_1, v_2\}$ y $\{w_1, w_2\}$ respectivamente; aplicando el producto tensorial se tiene como base:

$$\{v_1 \otimes w_1, v_1 \otimes w_2, v_2 \otimes w_1, v_2 \otimes w_2\}.$$

De la misma manera, consideremos dos qubits, cada uno con base $\{|0\rangle, |1\rangle\}$; realizando la misma operación se tiene:

$$\{|0\rangle, |1\rangle\} \otimes \{|0\rangle, |1\rangle\} = \{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}.$$

De manera más compacta $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Se obtienen así, las cuatro combinaciones superpuestas ($2^L = 4$) al mismo tiempo [2].

En general, la superposición de estados está dada por [4]:

$$\sum_{i=0}^{2^n-1} a_i |S_i\rangle,$$

donde a_i son números complejos. Puesto que $|a_i|^2$ es la probabilidad de medir u observar el estado $|S_i\rangle$ se cumple que $\sum_i |a_i|^2 = 1$.

2.2. Paralelismo cuántico

En muchas aplicaciones ejecutadas actualmente con computadoras, es necesario procesarlas a un gran velocidad, puesto que en muchos casos, es preciso obtener el resultado con un tiempo de respuesta lo más corto posible [5]. Actualmente, el procesamiento en paralelo es una forma eficaz de procesar la información, siendo los procesos paralelos los que se producen en diferentes recursos durante el mismo intervalo de tiempo [6]. Haciendo uso de la superposición de estados de los qubits, la computación cuántica ofrece un nuevo enfoque para ejecutar procesamiento en paralelo, donde no es necesario agregar procesadores a la máquina, solo es necesario un sistema de qubits es cido para poder obtener el procesamiento en paralelo (Fig. 3). A manera de ejem-

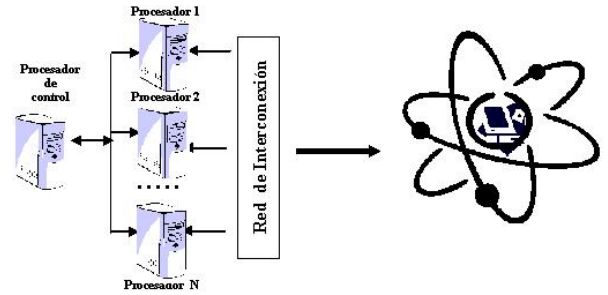


Figura 3. Arquitectura matricial (mod [5]) versus computación cuántica (arquitectura RMN).

plo, se puede considerar un bit al cual se le aplica una función booleana f :

$$f : x \rightarrow f(x)$$

Puesto que el bit sólo puede ser representado por uno de los dos estados (0 ó 1), es necesario aplicar la función f dos veces ó utilizar dos procesadores trabajando en paralelo.

$$f : 0 \rightarrow f(0)$$

$$f : 1 \rightarrow f(1)$$

A diferencia del bit, un qubit puede existir en una superposición de estados cuántico:

$$\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

y por linealidad de mecánica cuántica la función f se aplica una sola vez a los dos valores superpuestos 0 y 1.

$$\frac{1}{\sqrt{2}} |f(0)\rangle + \frac{1}{\sqrt{2}} |f(1)\rangle.$$

De manera similar, la Fig. 4 muestra cómo se aplica una función g a un registro de dos qubits, la cual tiene cuatro valores de entrada superpuestos con sus respectivas probabilidades c_0, c_1, c_2, c_3 . En este caso, la función g evalúa los cuatro valores superpuestos en una sola iteración, lo que su homóloga lo haría en cuatro pasos o utilizando cuatro procesadores en paralelo. Lo mismo sucede si se tiene L qubits en una superposición cuántica con 2^L valores de entrada; la función g se aplica a estos valores en un solo paso computacional, evitando así repetir el mismo proceso 2^L veces o utilizar 2^L diferentes procesadores trabajando paralelamente.[3]

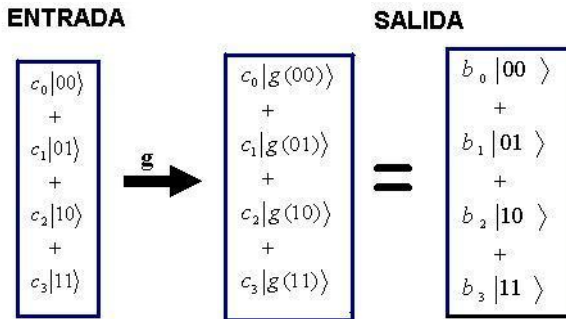


Figura 4. Paralelismo cuántico de un registro de dos qubits [3].

2.3. Compuertas cuánticas

Los estados superpuestos de un registro de qubits sólo pueden ser modificados mediante transformaciones unitarias las cuales son utilizadas como compuertas cuánticas. Existe una gran variedad de compuertas cuánticas cuya propiedad es la reversibilidad -se puede obtener el valor de entrada mediante el valor de salida- la cual hace que difieran en mucho de las compuertas clásicas . En este artículo, sólo se hará referencia a la *compuerta de Hadamard* y a la *transformada discreta de Fourier* ya que estas son utilizadas en el algoritmo de factorización cuántico.

2.3.1. La transformada de Hadamard (H)

La compuerta de Hadamard transforma un registro de qubits en una superposición coherente de estados, según [2]

$$H : \begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ |1\rangle &\rightarrow \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle). \end{aligned}$$

Aplicando H a n qubits individuales en estado $|0\rangle$, se crea una superposición de todos los 2^n estados posibles,

obteniendo así la representación binaria [2] de 0 a $2^n - 1$ ⁵

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle.$$

2.3.2. La Transformada Discreta de Fourier (TDF_q)

La TDF_q , es definida como una transformación unitaria en q dimensiones [7] la cual se denota por:

$$TDF_q : |a\rangle \rightarrow \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \exp\left(\frac{2\pi ac}{q}\right) |c\rangle$$

La TDF_q transforma el estado $|a\rangle$ a un nuevo estado $|c\rangle$, para algún a en el rango $0 \leq a < q$, donde el número de qubits de q es polinomial. Así, un estado con periodo r se transforma a otro con periodo q/r [7]

3. ALGORITMO DE FACTORIZACIÓN

El algoritmo de factorización ha llamado la atención de varios científicos debido a que los sistemas criptográficos basan su seguridad en la dificultad de factorizar números grandes.

En 1994, es publicado el artículo "Algorithms for quantum computation: discrete logarithms and factoring" [8], en el cual muestra un nuevo enfoque del algoritmo de factorización, donde se combinan ingeniosamente los principios de la mecánica cuántica con la teoría de números.

El algoritmo de factorización cuántico es fundamental en el avance de la computación cuántica, puesto que con el desarrollo del mismo, la investigación en este campo comenzó a tomar mayor importancia, debido a que este algoritmo pone en riesgo los sistemas criptográficos actuales. Considerando que es un algoritmo con un tiempo de ejecución polinomial, factorizaría un número de 1024 dígitos (número de dígitos que generalmente se usan en sistemas criptográficos) en 4.5 minutos, lo que a su análogo clásico le tomaría 300 millones de años [9].

A continuación se presenta un análisis en base a ejemplos del algoritmo de factorización cuántico con un previo análisis del algoritmo clásico.

3.1. El problema

El problema de factorización se puede resumir en que dado un número impar no primo, se deben encontrar los dos factores primos de N , $fac1$ y $fac2$ tales que:

$$N = fac1 \cdot fac2.$$

Para poder hallar estos factores, el algoritmo cuántico difiere en mucho del clásico; sin embargo, el análisis del algoritmo clásico permitirá comprender mejor el algoritmo cuántico.

⁵Para más detalles ver ejemplo 2.

3.2. Algoritmo clásico

Dado un número N impar no primo, este algoritmo consta de los siguientes pasos:

- (i) Seleccionar un $y < N$ tal que y sea coprimo de N , es decir, $\text{mcd}(y, N) = 1$.⁶
- (ii) Calcular el orden r de $y \bmod N$.
- (iii) Si r es par y $y^{r/2} \not\equiv -1 \pmod{N}$, entonces: $x = y^{r/2}$; caso contrario volver a (i).

1. Calcular los dos factores primos:
 $\text{fac1} = \text{mcd}(x + 1, N)$
 $\text{fac2} = \text{mcd}(x - 1, N)$.

La probabilidad de obtener con éxito los dos factores de N esta dada por[7]:

$$\text{Prob} \left(r \text{ sea par y } y^{r/2} \not\equiv \pm 1 \pmod{N} \right) \geq 1 - \frac{1}{2^{k-1}},$$

donde k es el número de factores primos de N .

A manera de ejemplo, tomemos $N = 55$ (impar no primo).

1. En este caso los valores de y coprimos a N son: $\{1, 2, 3, 4, 6, 7, 8, 9, 12, 13, \dots, 54\}$ de los cuales tomamos uno al azar, digamos, $y = 9$.
2. Debemos hallar el orden r de $9 \bmod 55$. Para esto, se hará uso de la siguiente definición de teoría de números.

“Suponiendo que el $\text{mcd}(y, N) = 1$, entonces el orden r de $y \bmod N$ es la menor potencia de y congruente a 1 mod N ($y^r \equiv 1 \pmod{N}$)” [7].

Puesto que $y = 9$ y sus potencias son:

$$\{9^1, 9^2, 9^3, 9^4, \dots, 9^{10}, \dots\} = \{9, 81, 729, 6561, \dots, 3486784401, \dots\}$$

y cuyos valores de congruencia están dadas por ($y^i \bmod 55$; $i = 1, 2, 3, \dots$):

$$\{9, 26, 14, 16, 34, 31, 4, 36, 49, 1, 9, 26, \dots\},$$

se puede ver que la menor potencia de y congruente a 1 mod 55 es 9^{10} , es decir $9^{10} \equiv 1 \pmod{55}$. En consecuencia $r = 10$. En otras palabras, el primer $y^i \bmod N = 1$ es $y^{10} \bmod 55$ de donde se obtiene $r = 10$

3. Dado que r es par y $9^{10/2} \not\equiv -1 \pmod{55}$, $x = 9^{10/2} = 59049$.

En el caso que r fuera par, pero $y^{r/2} \equiv -1 \pmod{55}$, el método falla, debido a que se obtienen factores triviales (1 ó -1); por lo que se tiene que volver a elegir un nuevo y ; de la misma manera, si r es impar.

⁶Para obtener eficientemente el mcd (máximo común denominador), se puede hacer uso del algoritmo de Euclides.

4. Lo que resta hallar, son los dos factores primos de 55, los cuales se obtienen de la siguiente manera:

$$\begin{aligned} x &= 59049 \\ \text{fac1} &= \text{mcd}(59050, 55) = 5 \\ \text{fac2} &= \text{mcd}(59048, 55) = 11, \end{aligned}$$

donde claramente se puede ver que 5 y 11 son los dos factores primos de 55.

3.3. Algoritmo cuántico de Shor

Basado en el algoritmo clásico de factorización, Peter Shor describe un algoritmo cuántico del cual halla el orden r de $y \bmod N$ en tiempo polinomial, es decir requiere $\text{poly}(\log N)$ pasos en su ejecución [7].

Aunque este algoritmo es relativamente complicado, es muy interesante para analizar el nuevo enfoque que la mecánica cuántica da la computación.

A continuación se presenta paso a paso dicho algoritmo, mostrándose las diferencias existentes entre el algoritmo clásico y cuántico, así como las características principales del algoritmo cuántico.

En primer lugar se hará uso de dos registros, uno de L qubits y otro de L' qubits de longitud

$$\Psi = |L\rangle |L'\rangle;$$

el primer registro permitirá determinar el orden r de $y \bmod N$ y el segundo servirá como auxiliar.

Paso 1. Hallar L y L' .

Para esto, se escoge un $q = 2^L$ ⁷ tal que $N^2 \leq q < 2 \cdot N^2$ [4]. L' se obtiene hallando la longitud de $N-1$ en binario [11].

Ejemplo 1. Para ver la diferencia con el algoritmo clásico, tomemos nuevamente $N = 55$ y $y = 9$.

Para obtener L , tomemos $q = 2^{12} = 4096$, valor que pertenece al rango $55^2 \leq q < 2 \cdot 55^2$, es decir, que la desigualdad $3025 \leq 4096 < 6050$ se cumple. Dado que $q = 2^{12}$, $L = 12$.⁸

El valor de L' debe ser capaz de almacenar de 0 a 54 en binario, entonces como $54 = 110110$, $L' = 6$.

Paso 2. Una vez obtenidos L y L' , se debe poner la máquina en una superposición de estados cuántica. Para esto, se deben preparar los registros L y L' en estado $|0\rangle$ y luego aplicar la transformada de Hadamard al primer registro, con lo cual se obtiene el siguiente estado de superposición [7].

⁷La elección de q como potencia de 2 es para asegurar que la $TDFq$ sea ejecutada eficientemente [7], es decir que se pueda aplicar en un paralelismo cuántico masivo.

⁸Para fines de la reproducción del algoritmo cuántico, el valor de L se obtuvo de la relación $L = \frac{\ln(2 \cdot N^2)}{\ln 2}$, considerando el rango al que pertenece q .

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |0\rangle, \quad (1)$$

donde a representa los números binarios de 0 a $q-1$.

Ejemplo 2. Para ver como se obtiene el estado de superposición (1), en primer lugar, consideremos un registro con solo dos qubits, los cuales están en estado $|0\rangle$ [10].

Sabemos que la compuerta de Hadamard (H) transforma cada qubit de la siguiente manera:

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

aplicando esta compuerta a los dos qubits, el registro cambia a un estado superpuesto de cuatro valores distintos, como se muestra a continuación:

$$\begin{aligned} & \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) = \\ & \frac{1}{\sqrt{2^2}} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \\ & \frac{1}{\sqrt{2^2}} \sum_{a=0}^{2^2-1} |a\rangle = \\ & \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle. \end{aligned}$$

Donde 00 es el binario de 0, 01 binario de 1, 10 de 2 y 11 de 3. (Para fines de ejemplificación se tomarán números enteros).

De la misma manera, se aplica H a los 12 qubits del primer registro del ejemplo 1.

$$\Psi = \underbrace{|000000000000\rangle}_{L=12 \text{ qubits}} \underbrace{|000000\rangle}_{L'=6 \text{ qubits}}$$

Obteniendo así, la superposición deseada con $2^{12} - 1$ valores superpuestos.

$$\begin{aligned} & \frac{1}{\sqrt{2^{12}}} (|0, 0\rangle + |1, 0\rangle + |2, 0\rangle + \dots + |4095, 0\rangle) = \\ & \frac{1}{\sqrt{2^{12}}} \sum_{a=0}^{2^{12}-1} |a\rangle |0\rangle. \end{aligned}$$

Paso 3. Calcular la función $y^a \bmod N$ para cada valor de a entre 0 y $q-1$ y almacenar este valor en el segundo registro:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |y^a \bmod N\rangle \quad (2)$$

A diferencia del algoritmo clásico, que calcula esta función de manera secuencial, el algoritmo cuántico lo hace en una sola iteración, en un paralelismo cuántico masivo, aplicando la función a cada valor de a al mismo tiempo. (Para más detalles ver Sec. 2).

Ejemplo 3. Teniendo en cuenta el estado del ejemplo 2,

$$\frac{1}{\sqrt{2^{12}}} (|0, 0\rangle + |1, 0\rangle + |2, 0\rangle + \dots + |10, 0\rangle + \dots + |4095, 0\rangle).$$

Vamos a calcular la función $9^a \bmod 55$ para cada valor de a de 0 hasta 4095.

$$\begin{aligned} & \frac{1}{\sqrt{2^{12}}} (|0, 9^0 \bmod 55\rangle + |1, 9^1 \bmod 55\rangle + \dots + \\ & |10, 9^{10} \bmod 55\rangle + \dots + |4095, 9^{4095} \bmod 55\rangle) \end{aligned}$$

teniendo como resultado:

$$\begin{aligned} & \frac{1}{\sqrt{4096}} (|0, 1\rangle + |1, 9\rangle + |2, 26\rangle + \dots + \\ & |9, 49\rangle + \dots + |10, 1\rangle + \dots + |4095, 34\rangle) = \\ & \frac{1}{\sqrt{4096}} \sum_{a=0}^{4095} |a\rangle |9^a \bmod 55\rangle. \end{aligned}$$

Se nota que los valores obtenidos en el segundo registro son los mismos que los valores de congruencia obtenidos en el algoritmo clásico; con la gran diferencia que los 4096 valores se obtienen en una sola iteración.

Observando el registro y siguiendo la lógica del algoritmo clásico, se diría que $r = 10$, obteniendo de esta manera el orden de la función. Sin embargo, si se quiere observar este resultado, por principios de mecánica cuántica, el estado (2) colapsa a un nuevo estado, en donde la información del orden r se encuentra inmersa. En este sentido, existe otra manera para obtener el resultado, la cual se explica a continuación.

Paso 4. Al querer obtener el orden r del estado (2), este produce un valor k que es el resultado de alguna potencia de $y \bmod N$, es decir, $k = y^{a_0} \bmod N$ para algún valor menor a_0 (cualquier potencia de $y \bmod N$ puede ser observada).

En consecuencia el nuevo estado colapsado estará dado por: [7].

$$|\Phi\rangle = \frac{1}{\sqrt{A+1}} \sum_{d=0}^A |a_0 + d \cdot r, k\rangle,$$

donde A es el entero más grande menor que $\frac{q-a_0}{r}$ y $a_0 \leq r$.

Para fines de cálculo, $M = A + 1$, teniendo como resultado:

$$\Phi = \frac{1}{\sqrt{M}} \sum_{d=0}^{M-1} |a_0 + d \cdot r, k\rangle \quad (3)$$

Donde M es la longitud de la serie:

$$a_0, a_0 + r, a_0 + 2 \cdot r, \dots, a_0 + (M - 1) \cdot r$$

y se puede obtener de la equivalencia $M \approx \frac{q}{r}$ [12].

Por otro lado, si r es el orden de y mod N , entonces se cumple que: [7].

$$y^{a_0} \equiv y^{a_0 + d \cdot r}$$

para todo d desde 0 hasta $M - 1$

Ejemplo 4. Continuando con nuestro ejemplo, veamos lo que sucede al querer observar r en el estado (2)

$$= \frac{1}{\sqrt{4096}} (|0, 1\rangle + |1, 9\rangle + |2, 26\rangle + |3, 14\rangle + |\mathbf{4, 16}\rangle + \dots + |10, 1\rangle + \dots + |4095, 34\rangle).$$

Suponiendo que se obtuvo $k = 16$ (aunque se puede obtener otro valor),

$$\Phi = \frac{1}{\sqrt{410}} (|4, 16\rangle + |14, 16\rangle + |24, 16\rangle + \dots + |4094, 16\rangle),$$

es decir,

$$\Phi = \frac{1}{\sqrt{410}} \sum_{d=0}^{410-1} |4 + d \cdot 10, 16\rangle.$$

Fácilmente se puede ver que $a_0 = 4$. Para verificar que $r = 10$ y $M = 410$, comprobemos que $y^{a_0} \equiv y^{a_0 + d \cdot r} \pmod{N}$, que no es más que verificar que para un r dado se cumple: $\text{mod}(y^{a_0}, N) = \text{mod}(y^{a_0 + d \cdot r}, N) \forall d$ entre 0 y M

es decir, verificar si el residuo de ambas potencias es el mismo. En este caso, se tiene:

$$\text{mod}(9^4, 55) = 16.$$

Entonces se debe comprobar si para un r dado la igualdad $\text{mod}(9^4, 55) = \text{mod}(9^{4+d \cdot r}, 55)$ se cumple. En la Tab. 1, se puede observar algunos ejemplos que se tomaron para obtener el valor de r . Como se puede ver, se realizaron los calculos pertinentes y se verificó que para valores de $r = 1, 2, \dots, 8, 9, 11, \dots$ la igualdad no se cumple, sin embargo, para $r = 10$ la igualdad es verdadera, en consecuencia se demostró que el r buscado es igual a 10. Finalmente, con $r = 10$ se tiene:

$$M = \frac{q}{r} = \frac{4096}{10} \approx 410$$

Paso 5. Transformada Discreta de Fourier. Una vez obtenido el estado (3), lo que se debe hacer es obtener r (valor inmerso en este estado). Para esto, Peter Shor propuso aplicar la TDF_q al estado (3), siendo el corazón de este algoritmo, ya que permite obtener la periodicidad (el orden r) a partir de una distribución de probabilidades.

Como se vió en Sec.2, la TDF_q transforma un estado $|a\rangle$ a un nuevo estado $|c\rangle$,

$$TDF_q : |a\rangle \rightarrow \frac{1}{q^{1/2}} \sum_{c=0}^{q-1} \exp\left(\frac{2\pi i a c}{q}\right) |c\rangle.$$

Aplicando la TDF_q al estado (3) se tiene:

$$TDF_q |\Phi\rangle = \frac{1}{q^{1/2}} \sum_{c=0}^{q-1} \frac{1}{\sqrt{M}} \sum_{d=0}^{M-1} \exp\left(\frac{2\pi i (a_0 + d r) c}{q}\right) |c, k\rangle$$

y realizando algunas operaciones en función de c y d ,

$$TDF_q |\Phi\rangle = \frac{1}{\sqrt{qM}} \sum_{c=0}^{q-1} \sum_{d=0}^{M-1} \exp\left(\frac{2\pi i (a_0 + d r) c}{q}\right) |c, k\rangle$$

y

$$TDF_q |\Phi\rangle = \sum_{c=0}^{q-1} \frac{\exp(2\pi i c \cdot a_0 / q)}{\sqrt{qM}} \sum_{d=0}^{M-1} \exp\left(2\pi i \frac{c d r}{q}\right) |c, k\rangle.$$

Finalmente se tiene:

$$TDF_q |\Phi\rangle = \sum_{c=0}^{q-1} \frac{e^{2\pi i c \cdot a_0 / q}}{\sqrt{qM}} \left(\sum_{d=0}^{M-1} \zeta^d \right) |c, k\rangle$$

con $\zeta = e^{2\pi i c / q}$ [12].

Ejemplo 5. Tomando en cuenta el análisis del Paso 5 y aplicando la TDF_q al estado (3) del Ej. 4,

$$\Phi = \frac{1}{\sqrt{410}} \sum_{d=0}^{410-1} |4 + d \cdot 10, 16\rangle,$$

se tiene como resultado:

$$TDF_q |\Phi\rangle = \sum_{c=0}^{4095} \frac{e^{2\pi i c \cdot 4 / 4096}}{\sqrt{4096 \cdot 410}} \left(\sum_{d=0}^{409} \zeta^d \right) |c, 16\rangle$$

con $\zeta = e^{2\pi i c \cdot 10 / 4096}$

Paso 6. Al aplicar la TDF_q al estado (3), el nuevo estado estará gobernado por una distribución de probabilidades, la cual esta dada por:[7]

$$\text{Prob}(c) = \frac{r}{q^2} \left| \sum_{d=0}^{q/r-1} \exp\left(\frac{2 \cdot \pi \cdot d \cdot (r \cdot c \pmod{q})}{q}\right) \right|^2$$

Con $M = \frac{q}{r}$ se tiene:

$$\text{Prob}(c) = \frac{1}{q \cdot M} \left| \sum_{d=0}^{M-1} \exp\left(\frac{2 \cdot \pi \cdot d \cdot (r \cdot c \pmod{q})}{q}\right) \right|^2 \quad (4)$$

TABLA 1

Valores obtenidos que comprueban que $r = 10$ es el orden de $y \bmod N$.

$\text{mod}(9^{4+d \cdot r}, 55)$	d=1	d=2	d=3	d=4	d=5	...	d=409
$r = 1$	34	31	4	36	49	...	14
$r = 2$	31	36	1	26	16	...	26
...
$r = 8$	26	1	36	31	16	...	31
$r = 9$	14	26	9	1	49	...	34
$r=10$	16	16	16	16	16	...	16
$r = 11$	34	31	4	36	49	...	14
...

Donde $\text{Prob}(c)$ es la probabilidad de obtener cualquier valor de c entre 0 y $q - 1$.

Dado que existe ciertos valores de c los cuales tienen una mayor probabilidad de ser observados, estos son cercanos a los múltiplos de q/r y cumplen con la relación:[8]

$$-\frac{r}{2} \leq r \cdot c \bmod q \leq \frac{r}{2} \quad (5)$$

Hay precisamente r valores de $c \bmod q$ que satisfacen la ecuación y la probabilidad de ver un estado c será de al menos $1/3 \cdot r^2$. [8]

Ejemplo 6. La Fig.5 muestra la distribución de probabilidades para nuestro ejemplo. Donde los valores de c y sus probabilidades son:

c : 0, 410, 819, 1229, 1638, 2048, 2458, 2867, 3277, 3686.

$\text{prob}(c)$: 0,100, 0,057, 0,087, 0,087, 0,057, 0,100, 0,057, 0,087, 0,087, 0,057.

Como se podrá observar, los valores de c con mayor probabilidad son cercanos a los múltiplos de 410 ($\frac{q}{r} = 409,6$) y hay exactamente 10 ($r = 10$) valores de c que cumplen con la relación (5).

A manera de ejemplo, veamos si $-\frac{r}{2} \leq r \cdot c \bmod q \leq \frac{r}{2}$ se cumple para $c=1638$ y $c=2458$.

En el caso de $c=1638$ se tiene:

$$-\frac{10}{2} \leq 10 \cdot 1638 \bmod 4096 \leq \frac{10}{2}$$

$$-5 \leq -4 < 5.$$

En este caso, se tiene un valor negativo (-4), con esto se puede notar que la relación de desigualdad hace referencia a valores congruentes tanto negativos como positivos.

Para $c=2458$ se tiene:

Distribución de Probabilidades

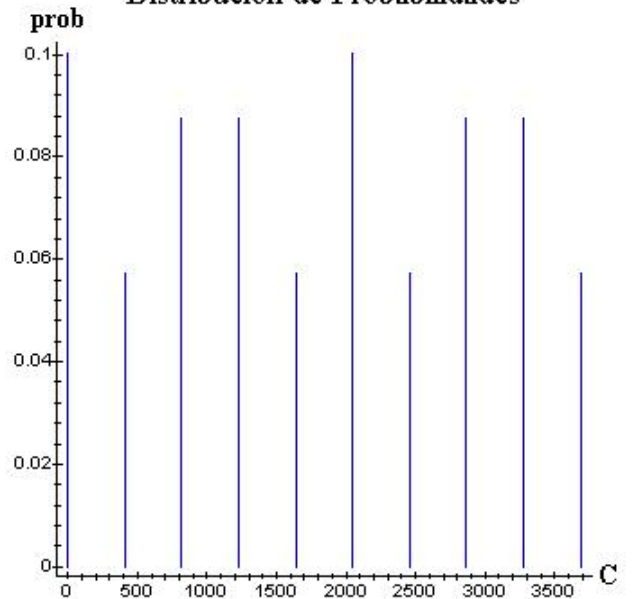


Figura 5. Distribución de probabilidades para $N=55$, con $q = 4096$ y $r = 10$.

$$-\frac{10}{2} \leq 10 \cdot 2458 \bmod 4096 \leq \frac{10}{2}$$

$$-5 \leq 4 < 5.$$

En ambos casos se comprueba la desigualdad.

Paso 7. Una vez obtenido los valores de c , se escoge uno aleatoriamente y se halla un valor d el cual debe satisfacer la relación:

$$-\frac{1}{2q} \leq \frac{c}{q} - \frac{d}{r} \leq \frac{1}{1q}$$

Para algún d entre $0 \leq d \leq r - 1$.

La fracción $\frac{d}{r}$ puede ser hallada eficientemente usando la expansión de fracciones continuas de $\frac{c}{q}$ como uno de sus convergentes.

Ejemplo.7 Para nuestro ejemplo, tomemos $c=2458$ para hallar $\frac{d}{r}$ mediante la expansión de fracciones continuas.

$$\frac{c}{q} = \frac{2458}{4096} = \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{409}}}}}$$

cuyas convergentes son:

$$\begin{aligned} \frac{1}{1} &= 1 \\ \frac{1}{1 + \frac{1}{1}} &= \frac{1}{2} \\ \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}} &= \frac{2}{3} \\ \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}}} &= \frac{3}{5} \\ \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{409}}}}} &= \frac{1229}{2048} \end{aligned}$$

De aquí se tiene que $\frac{d}{r} = \frac{3}{5}$, ya que el denominador no excede a 55 ($N = 55$).

El orden r de $y \bmod N$ es un múltiplo de $r = 5$. La Tab. 2 muestra como se comprueba la igualdad $y^a \equiv 1$,

TABLA 2

Múltiplos de 5 con los cuales se comprueba que $r = 10$.

a	$y^a \bmod N = 9^a \bmod 55$
5	34
10	1
15	34

donde a es un múltiplo de 55. Obteniendo así el orden r , el cual nos permitirá obtener los dos factores primos de 55.

$$\begin{aligned} x &= y^{r/2} = 9^{10/2} = 59049 \\ \text{fac1} &= \text{mcd}(59050, 55) = 5 \\ \text{fac2} &= \text{mcd}(59048, 55) = 11 \end{aligned}$$

3.4. Tiempos de ejecución

Una forma de comparar los resultados que se obtienen utilizando ambos algoritmos, es obtener los tiempos de ejecución [13]: estos se muestran en las Figs. 6 y 7 para los casos clásico y cuántico respectivamente.

Como se puede observar en la Fig 6, en promedio, el tiempo de ejecución crece con el número N a factorizar.

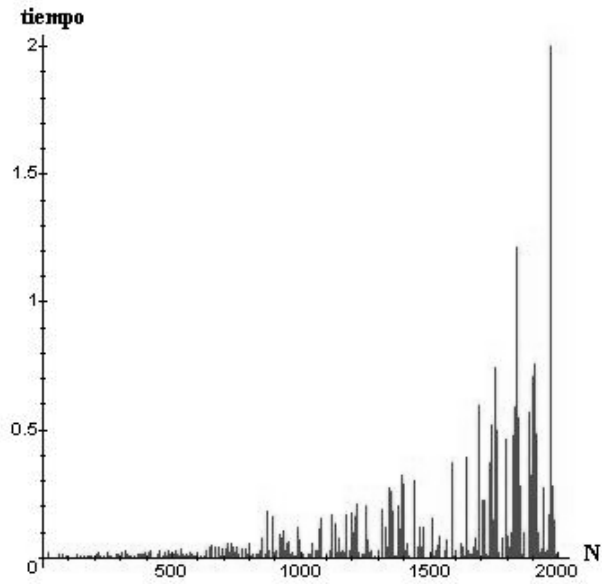


Figura 6. Tiempo de ejecución para la factorización de números impares no primos entre 15 y 2000, utilizando el algoritmo clásico.

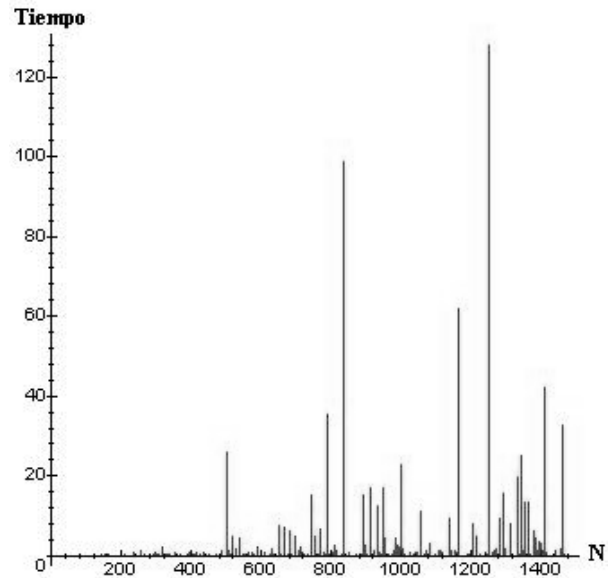


Figura 7. Tiempo de ejecución de la pseudo-simulación del algoritmo cuántico para los números impares no primos entre 15 y 1500.

Similar comportamiento (salvo algunos casos patológicos) se tiene en la Fig. 7, con la diferencia de que los tiempos de ejecución son mucho mayores. Lo último, no debe sorprendernos, puesto que para la aplicación del algoritmo cuántico se tuvo que hacer una pseudo-simulación (muchos más pasos en el programa) y además, que se trabajó con una computadora clásica⁹.

⁹Los resultados fueron obtenidos utilizando una computadora con 224 Mb de memoria RAM con un procesador de 1.7 GHz de velocidad

4. CONCLUSIONES Y PERSPECTIVAS

La importancia de este trabajo radica en el hecho de haber demostrado que es posible implementar un algoritmo cuántico de factorización. Si bien, los tiempos de ejecución son mucho mayores a los que se tienen trabajando con el algoritmo clásico, esto no refleja la verdadera potencialidad del algoritmo cuántico debido a que no se contó con una computadora cuántica. De aplicarse el algoritmo de Shor en una computadora cuántica, se prevé una reducción exponencial del tiempo de procesamiento, lo que proporciona una forma nueva de enfocar el problema de factorización. En base a lo anterior, se puede afirmar que la aplicación de este algoritmo podría poner en riesgo los sistemas criptográficos actuales. Por otro lado, la aplicación de la computación cuántica abre posibilidades antes no imaginadas, como el manejo de bits superpuestos, procesamiento de información en paralelo sin necesidad de procesadores adicionales, lo que trae consigo una reducción exponencial de recursos computacionales.

REFERENCIAS

- [1] M.L. Steffen, L.M.K. Vandersypen, I.L. Chuang, Toward quantum computation: a five-qubit quantum processor. *IEEE*, (2001) 24–34.
- [2] E. Rieffel, W. Polak. An introduction to quantum computing for non-physicists, *ACM Computing Surveys (CSUR)*, **32** 3, (2000) 300–335.
- [3] A. Barenco, A. Ekert, A. Sanpera, C. Machiavello, A short introduction to quantum computation. (1996) <http://www.qubit.org/index.html>, Acceso: marzo, 2005
- [4] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing* **26** 5 (1997) 1484–1509.
- [5] E. Alcalde, F. Ormachea, J. Portillo, F. Garcia Mera-yo. *Arquitectura de Ordenadores*. McGraw Hill Interamericana de España S.A., (1991).
- [6] H. Hwang, F.A. Briggs, *Arquitectura de computadoras y procesamiento en paralelo*. McGraw-Hill, (1998)
- [7] A. Ekert, R. Jozsa. Quantum computation and Shor's factoring algorithm. *Review of modern Physics*, **68** 3, (1996) 733–753.
- [8] P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring. *Proc 35th Ann. Symp. on found of Comp. Sci.*, (1994) 124–134.
- [9] R.J. Hughes, Cryptography, quantum computation and trapped ions, *arXiv.quant-ph/9712054*, (1997).
- [10] A. Ekert, *Quantum cryptoanalysis-Introduction*, (1995) <http://www.qubit.org/index.html>, Acceso: marzo, 2005
- [11] J.F Schneiderman, M.E. Stanley, P.K. Aravind, *A pseudo-simulation of Shor's quantum factoring algorithm*. Department of Physics, Worcester Polytechnic Institute.
- [12] G.E. Moorhouse, U.W. Math, *Shor's algorithm for Factoring Large Integer*
- [13] C.L. Mayda, *Efectos de la computación cuántica en la tecnología*, Tesis de grado Carrera Informática, Universidad Mayor de San Andrés, (2004).