

# Diseño y simulación de un procesador cuántico superconductor

Miguel Casanova  
Departamento de Electrónica y Circuitos<sup>1</sup>, Universidad Simón Bolívar

2018  
September

<sup>1</sup>I am no longer a member of this department

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. JUSTIFICACION	6
1.2. OBJETIVOS	8
1.2.1. Objetivo General:	8
1.2.2. Objetivos Específicos:	8
1.2.3. Fases del Proyecto	8
1.2.4. REFERENCIAS	9
<b>2. Información cuántica</b>	<b>12</b>
2.1. Función de onda	12
2.2. Espacio de Hilbert	13
2.3. Delta de Kronecker	14
2.4. Operadores hermíticos	14
2.5. Operadores unitarios	15
2.6. Notación de Dirac	15
2.7. Producto tensorial	17
2.8. Postulados de la mecánica cuántica	18
2.9. Matriz densidad	19
2.10. Traza parcial	21
2.10.1. Comparación con el producto tensorial	22
2.11. Entrelazamiento	22
2.12. Computación cuántica	23
2.12.1. Qubits	23
2.12.2. Esfera de Bloch	23
2.12.3. Conmutador y anticonmutador	24
2.12.4. Matrices de Pauli	25
2.12.5. Circuitos cuánticos	25
2.12.6. Compuertas cuánticas de un qubit	28
2.12.7. Compuertas multiqubit	34

2.12.8. Conjuntos universales de compuertas cuánticas . . . . .	37
2.12.9. Criterios de DiVincenzo . . . . .	38
2.13. Fidelidad . . . . .	38
2.14. Medidas proyectivas . . . . .	39
<b>3. Superconductividad</b>	<b>41</b>
3.1. Cuantización macroscópica y superconductividad . . . . .	41
3.2. La teoría BCS . . . . .	43
3.3. Cuantización del flujo magnético y efecto tunel Giaever . . . . .	51
3.4. Efecto Josephson . . . . .	57
3.5. Componentes de la corriente en las uniones de Josephson . . . . .	62
3.6. Qubits superconductores . . . . .	63
3.7. Arquetipos de qubits superconductores . . . . .	64
3.7.1. Qubit de carga . . . . .	64
3.7.2. Qubit de flujo . . . . .	64
3.7.3. Qubit de fase . . . . .	64
3.8. Transmones . . . . .	65
3.9. Hamiltonianos multiqubit de transmones . . . . .	65
3.9.1. Acoplamiento capacitivo . . . . .	66
3.9.2. Acoplamiento por el resonador . . . . .	66
3.9.3. Acoplamiento de JJ . . . . .	66
3.9.4. Acoplamiento afinable/calibrable . . . . .	66
3.10. Compuertas cuánticas en transmones . . . . .	66
3.10.1. El operador de evolución temporal . . . . .	66
3.10.2. Pulsos de microondas . . . . .	67
3.10.3. Régimen rotacional del pulso . . . . .	67
3.10.4. Efecto del pulso sobre el qubit . . . . .	68
3.10.5. Régimen dispersivo . . . . .	68
3.10.6. Rotaciones X-Y . . . . .	68
3.10.7. Compuerta de entrelazamiento . . . . .	69
3.10.8. Compuertas compuestas . . . . .	69
<b>4. El simulador</b>	<b>70</b>
4.1. Parámetros de los sistemas simulados . . . . .	71
4.2. Compuertas simples . . . . .	71
4.2.1. Rx y Ry . . . . .	72
4.2.2. iSWAP . . . . .	73
4.3. Compuertas compuestas . . . . .	75
4.3.1. X . . . . .	75
4.3.2. Y . . . . .	75

4.3.3.	Rz	75
4.3.4.	Z	76
4.3.5.	H	76
4.3.6.	CNOT	76
4.3.7.	SWAP	76
4.3.8.	Compuertas condicionales generales	76
4.3.9.	CP	78
<b>5.</b>	<b>Algoritmo de Grover</b>	<b>82</b>
5.1.	El algoritmo	87
5.2.	Simulación	87
<b>6.</b>	<b>Algoritmo de Shor</b>	<b>89</b>
6.1.	Transformada cuántica de Fourier	89
6.2.	Estimación de fase	91
6.3.	Estimación de orden	93
6.4.	Expansión en fracciones continuas	96
6.5.	Algoritmo de factorización de Shor	97
6.6.	Simulación	97
<b>7.</b>	<b>Google PageRank</b>	<b>99</b>
7.1.	El algoritmo de remiendo (parcheo) general	102
7.2.	Interpretación como una caminata aleatoria	103
7.3.	Cuantizando las caminatas aleatorias	104
7.4.	Caminata cuántica de Szegedy	104
7.5.	PageRank cuántico	106
7.6.	Circuitos de las caminatas cuánticas de Szegedy	106
7.7.	Simulaciones	111
7.7.1.	Grafo estrella	111
7.7.2.	Grafo corona	112
7.7.3.	Grafo arbol	113
7.7.4.	Grafo aleatorio	114
<b>A.</b>	<b>Cálculos de Hamiltonianos</b>	<b>116</b>
A.1.	Hamiltoniano de Jaynes-Cummings	116
A.2.	Hamiltoniano multiqubit	116
A.3.	Pulsos de microondas	116
A.4.	Régimen rotacional del pulso	117
A.5.	Efecto del pulso sobre el qubit	121
A.6.	Régimen dispersivo	122

A.7. Rotaciones X-Y . . . . .	125
A.8. Compuerta de entrelazamiento . . . . .	125
<b>B. Cálculos de matrices de adyacencia</b>	<b>126</b>
<b>C. Circuitos cuánticos</b>	<b>127</b>
<b>D. Códigos del simulador</b>	<b>136</b>
D.1. Wolfram Mathematica . . . . .	136
D.2. Python . . . . .	144
<b>E. Códigos de la simulación del algoritmo de Grover</b>	<b>156</b>
E.1. Wolfram Mathematica . . . . .	156
E.2. Python . . . . .	158
<b>F. Códigos de la simulación del algoritmo de Shor</b>	<b>160</b>
F.1. Wolfram Mathematica . . . . .	160
F.2. Python . . . . .	164
<b>G. Códigos de la simulación del algoritmo de PageRank</b>	<b>168</b>
G.1. Wolfram Mathematica . . . . .	168
G.2. Python . . . . .	168

# Índice de figuras

2.1. Esfera de Bloch . . . . .	24
2.2. Compuerta I en la esfera de Bloch . . . . .	28
2.3. Compuerta X en la esfera de Bloch . . . . .	29
2.4. Compuerta Z en la esfera de Bloch . . . . .	29
2.5. Compuerta Y en la esfera de Bloch . . . . .	30
2.6. Compuerta H en la esfera de Bloch . . . . .	31
2.7. Compuerta S en la esfera de Bloch . . . . .	31
2.8. Compuerta T en la esfera de Bloch . . . . .	32
2.9. Compuerta P en la esfera de Bloch . . . . .	32
2.10. Compuertas Rx, Ry y Rz en la esfera de Bloch . . . . .	33
3.1. Diagrama de Feynman de la interacción electrón-fonón-electrón	47
3.2. Construcción geométrica de los posibles electrones candidatos para formar pares de Cooper, siendo $\hbar K$ el momentum del centro de masas. . . . .	48
3.3. Cuantización del flujo magnético . . . . .	53
3.4. Imposibilidad de efecto túnel a través de la barrera . . . . .	55
3.5. Posibilidad de efecto túnel a través de la barrera . . . . .	56
3.6. Efecto Giaver: Efecto túnel entre un metal y un superconductor	57
3.7. Curva característica de una unión Josephson . . . . .	61
4.1. Rotaciones en X e Y de $2\pi$ . . . . .	72
4.2. Rotaciones en X e Y de $\pi$ . . . . .	73
4.3. Rotaciones en X e Y de $\frac{\pi}{2}$ . . . . .	73
4.4. Compuertas iSWAP y $\sqrt{iSWAP}$ aplicadas a $ 00\rangle$ . . . . .	74
4.5. Compuertas iSWAP y $\sqrt{iSWAP}$ aplicadas a $ 01\rangle$ . . . . .	74
4.6. Compuertas iSWAP y $\sqrt{iSWAP}$ aplicadas a $\frac{ 00\rangle+ 11\rangle}{\sqrt{2}}$ . . . . .	74
4.7. Compuertas iSWAP y $\sqrt{iSWAP}$ aplicadas a $\frac{ 0\rangle+ 1\rangle}{\sqrt{2}} \otimes \frac{ 0\rangle+ 1\rangle}{\sqrt{2}}$ .	74
5.1. Circuito del algoritmo de Grover, $k_{max}$ desconocido. . . . .	84

5.2.	Interpretación geométrica del operador difusión . . . . .	86
5.3.	Circuito del algoritmo de Grover. . . . .	87
5.4.	Evolución de las probabilidades en el algoritmo de Grover sin pérdidas . . . . .	88
5.5.	Evolución de las probabilidades en el algoritmo de Grover con pérdidas . . . . .	88
7.1.	Transformación de un grafo al crear la matriz de Google con $\alpha = \frac{1}{2}$ . . . . .	103
7.2.	Operador de permutación . . . . .	108
7.3.	Circuito de Loke para las caminatas cuánticas de Szegedy . .	108
7.4.	Grafo estrella . . . . .	111
7.5.	Grafo estrella . . . . .	112
7.6.	Grafo estrella . . . . .	113
7.7.	Grafo estrella . . . . .	114

# Índice de cuadros



## Capítulo 7

# Google PageRank

El algoritmo de PageRank fue desarrollado en 1996 en la Universidad de Stanford por Larry Page y Sergey Brin, los cuales fueron los fundadores de Google.

Este algoritmo se basa en la idea de que sitios web importantes tienen muchos vínculos que apuntan hacia ellos, lo que conduce a pensar en la web como una red ponderada orientada.

Existen muchos otros algoritmos, algunos más eficientes, pero la importancia de PageRank se sustenta en el poder económico de Google.

Ilustraremos el algoritmo de PageRank con un ejemplo sencillo:

Ejemplo:

Consideremos 5 páginas web distintas a las que denotaremos por 1, 2, 3, 4, y 5, y cuyo grafo es:

Pasos:

1. Determinar la matriz de adyacencia. Algunos autores denotan la matriz de de adyacencia por  $M$  en el protocolo de PageRank

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

2. Sumamos los elementos de cada una de las columnas.

$$\begin{matrix} 3 & 1 & 2 & 2 & 3 \end{matrix}$$

Estas sumas representan el número de links que salen del nodo o vértice de la página  $p_j$ , es decir:

$I(p_j) \equiv$  Importancia de la página  $j$

$\text{outdeg}(p_j) \equiv$  número de links que salen de la página  $p_j$

$$I(p_i) \equiv \sum_{j \in B_i} \frac{I(p_j)}{\text{outdeg}(p_j)}$$

$B_i \equiv$  conjunto de páginas que son linkeadas

3. Dividimos cada elemento de  $M$  por la suma de los elementos de la columna a la cual corresponde y llamaremos a la nueva matriz obtenida  $M'$

$$M' = \begin{pmatrix} 0 & 0 & 1/2 & 1/2 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 0 & 0 & 1/2 & 1/3 \\ 1/3 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1 & 0 & 0 & 0 \end{pmatrix}$$

4. El siguiente paso es encontrar un vector  $\vec{v}$  (algunos autores lo llaman  $\vec{I}$ ) que represente el PageRank de cada una de las páginas. Como tenemos 5 páginas web le asignamos a  $\vec{v}$  como valores  $\vec{v} = (a, b, c, d, e)^T$ , obteniendo así un vector de dimensión  $d = 5$ .
5. Obtenemos los valores de  $\{v_i\}$  a partir de los autovalores de  $M'$ , tal que:

$$M' \vec{v} = \lambda \vec{v} \text{ con } \lambda \in R$$

6. Determinamos los autovalores de  $M'$

$$\lambda_1 = 1; \quad \lambda_2 = \frac{-2}{3}; \quad \lambda_3 = \frac{-1}{2}; \quad \lambda_4 = \frac{-1}{3}; \quad \lambda_5 = \frac{1}{3}$$

Tomaremos sólo  $\lambda = 1 \rightarrow M'\vec{v} = \vec{v}$  (Ecuación autoconsistente)

7. Hallamos el autovector asociado a  $\lambda = 1$ . Obteniendo:

$$a = 6; \quad b = 1; \quad c = \frac{16}{3}; \quad d = \frac{14}{3}; \quad e = 3$$

8. Finalmente, Google ordena de mayor a menor las componentes de  $\vec{v}$ , quedándonos:

$$\begin{array}{rcl} & - & a \\ & - & c \\ \text{Pantalla} & \rightarrow & - \quad d \\ & - & e \\ & - & b \end{array}$$

La idea de PageRank de Google es que la importancia de una página viene dada por la cantidad de páginas que se enlazan con ella.

Surgen varios problemas:

1. Las matrices hyperlink (hiperenlace) pueden tener billones de entradas en filas y columnas.
2. Calcular los autovectores es un absurdo computacional.
3. Los estudios muestran que un nodo (página web) tiene un promedio de 10 enlaces, y las demás entradas de la matriz son cero.
4. No se encuentra  $\lambda = 1$  en la mayoría de los casos.

Por esta razón, un remedio (Patching) del algoritmo de PageRank fue el método de las potencias, en el cual la matriz hiperenlace

$$H_{ij} \equiv \begin{cases} \frac{1}{\text{outdeg}(P_j)} & \text{si } P_j \in B_i \\ 0 & \text{en otro caso} \end{cases}$$

debería converger a una solución autoconsistente

$$I^{k+1} = HI^k$$

donde se toma un vector  $I^0$  y se hace interactuar unas 100 veces y el orden mostrado de las páginas es el de  $I^{100}$ , ordenadas de mayor a menor. Si se normalizan las columnas de la matriz hipervínculo (hiperenlace)  $H$ , obtenemos otra matriz hiperenlace normalizada  $E$ .

**La matriz E:** se sabe de la teoría de matrices estocásticas que 1 es uno de sus autovalores. Además, también se sabe que la convergencia de  $I^k = EI^{k-1}$  a  $I = EI$  depende del segundo autovalor de  $\lambda_2$  de  $E$  y es un hecho que  $I^k = EI^{k-1}$  converge rápidamente si  $|\lambda_2|$  es cercano a cero.

## 7.1. El algoritmo de remiendo (parcheo) general

Asumamos que el caminante recorre el grafo siguiendo la web con una matriz estocástica  $E$  con probabilidad  $\alpha$ , y con probabilidad  $1 - \alpha$  podrá ir a cualquier página al azar que sea de su interés. La matriz web de este proceso será:

$$G \equiv \alpha E + \frac{1 - \alpha}{N} \mathbb{I} \text{Matriz de Google}$$

$\mathbb{I}$  es una matriz en la cual todas las entradas están establecidas en 1, y  $N$  el número de nodos.

Propiedades de  $G$ :

1. Es estocástica
2. Irreducible
3. Primitiva
4. El resultado de determinar el estado auto-consistente no depende del vector Google inicial  $I^0$

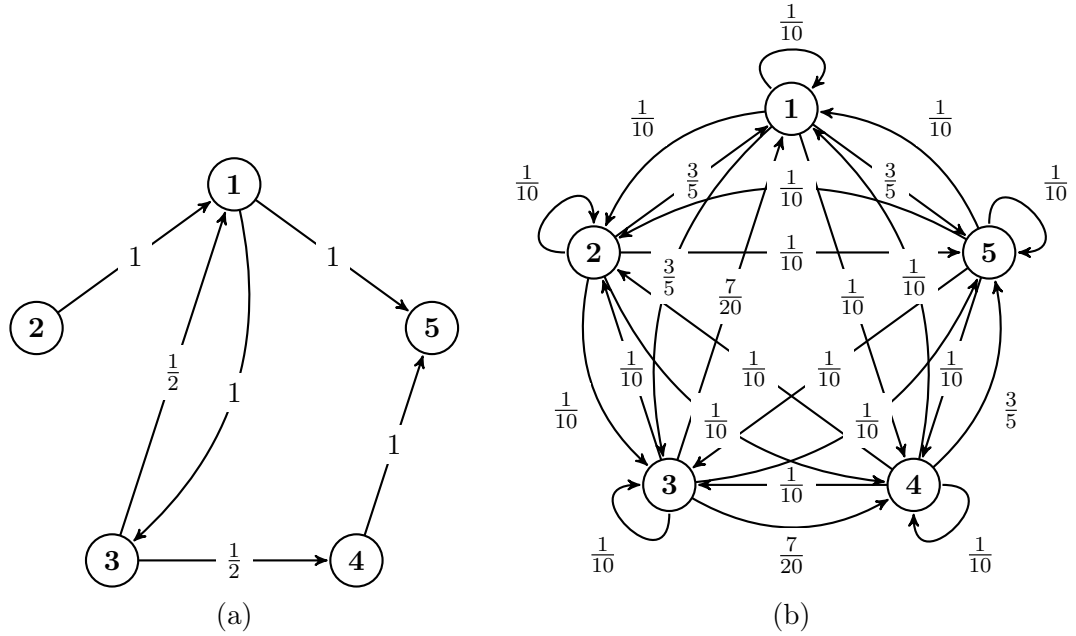


Figura 7.1: Transformación de un grafo al crear la matriz de Google con  $\alpha = \frac{1}{2}$ : Grafo correspondiente a la matriz de adyacencia (a) de la red E (b) remendada de Google G con  $\alpha = \frac{1}{2}$

## 7.2. Interpretación como una caminata aleatoria

La asignación de valores de importancia se puede replantear como la probabilidad de encontrar un caminante aleatorio en cierto nodo del grafo.

Del proceso:

De la ley de probabilidad total:

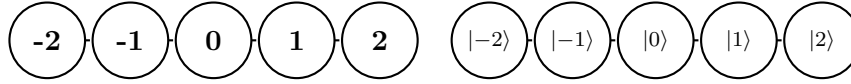
$$\begin{aligned}
 Pr(x^{(n+1)} = p_i) &= \sum_j G_{ij} Pr(x^{(n)} = p_j) \\
 Pr(x^{(n+1)} = p_i) &= \sum_j Pr(x^{(n+1)} = p_i | x^{(n)} = p_j) Pr(x^{(n)} = p_j) \\
 \implies G_{ij} &= Pr(x^{(n+1)} = p_i | x^{(n)} = p_j)
 \end{aligned}$$

En el contexto del Internet,  $G_{ij}$  es la probabilidad de que cierto internauta, que se encuentra en la página  $p_i$ , entre en la página  $p_j$ . El factor  $\alpha E_{ij}$  es la probabilidad de que lo haga presionando un enlace presente en  $p_i$ , mientras que  $\frac{1-\alpha}{N} \mathbb{I}$  es la probabilidad de que lo haga introduciendo la URL directamente.

El factor de amortiguamiento es libre y debe ser calibrado. Se suele usar  $\alpha = 0,85$

### 7.3. Cuantizando las caminatas aleatorias

La forma obvia y directa de cuantizar una caminata aleatoria sería sustituir el conjunto de nodos  $\{p_i\}$  por el conjunto de kets  $\{|i\rangle\}$ . Sin embargo, esto lleva a sistemas con operadores no unitarios y no es realizable.



Esto nos obliga a buscar maneras alternativas de cuantizar las caminatas aleatorias. La cadena anterior se podría cuantizar agregando un espacio "moneda." al espacio de Hilbert generado por  $\{|i\rangle\}$ . En este caso, el operador de difusión se interpreta como "lanzar la moneda" para decidir en qué dirección ir.

$$\begin{aligned} U &= \sqrt{p} |i+1\rangle\langle i| \otimes |c\rangle\langle c| + \sqrt{1-p} |i-p\rangle\langle i| \otimes |s\rangle\langle s| \\ U^\dagger &= \sqrt{p} |i\rangle\langle i+1| \otimes |c\rangle\langle c| + \sqrt{1-p} |i\rangle\langle i-p| \otimes |s\rangle\langle s| \\ UU^\dagger &= p |i+1\rangle\langle i+1| \otimes |c\rangle\langle c| + (1-p) |i-p\rangle\langle i-p| \otimes |s\rangle\langle s| \end{aligned}$$

Al realizar la suma sobre  $i$  se tiene  $\mathbb{1}$ , como se deseaba. Sin embargo, esta solución todavía no es satisfactoria, pues exige que  $p_{ij} = \frac{1}{\text{outdeg}(j)}$  para que  $UU^\dagger = \mathbb{1}$ .

Casi todas las cuantizaciones cometen estos dos pecados, aumentar la dimensión del espacio de Hilbert e imponer condiciones sobre el grafo; y en general, se debe cometer al menos uno de los dos.

**Nota:** También existen caminatas cuánticas continuas, no sólo discretas.

### 7.4. Caminata cuántica de Szegedy

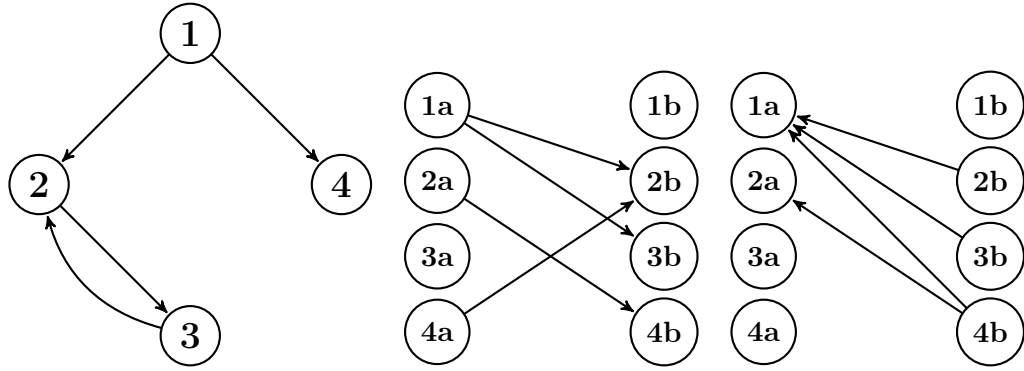
Existe un tipo particular de caminatas aleatorias conocido como caminatas bipartitas. En éstas se tiene dos conjuntos de nodos y sólo ocurren transiciones entre los dos conjuntos, no dentro del mismo.

Szegedy desarrolló una cuantización de estas caminatas. Para esto utilizó operadores de reflexión ( $W = \mathbb{1} - 2|w\rangle\langle w|$ , similares a los utilizados en el

algoritmo de Grover). Aprovechándose del hecho de que un par de reflexiones equivale a una rotación (como en el algoritmo de Grover), creó el siguiente operador de evolución de la caminata:  $U = (\mathbb{1} - 2B)(\mathbb{1} - 2A)$ , donde A es el proyector sobre las transiciones de la primera partición a la segunda y B de la segunda a la primera.

$$\begin{aligned} |\psi_i\rangle &= |i\rangle_1 \otimes \sum_j \sqrt{p_{ji}} |j\rangle_2 & |\psi_i\rangle &= \sum_i \sqrt{p_{ij}} |i\rangle_1 \otimes |i\rangle_2 \\ A &= \sum_i |\psi_i\rangle\langle\psi_i| & B &= \sum_j |\phi_j\rangle\langle\phi_j| \end{aligned}$$

Si tomamos un grafo cualquiera y lo duplicamos en la forma de un grafo bipartito con ambas particiones iguales y transiciones iguales en ambos sentidos, podemos cuantizar cualquier tipo de caminata. Sólo hay que pagar el precio de duplicar el espacio de Hilbert generado por  $\{|i\rangle\}$ :  $\mathcal{H}' = \mathcal{H} \otimes \mathcal{H}$ .



En estos casos, podemos escribir el operador de difusión en términos de sólo A, pues como la segunda partición es un reflejo de la primera,  $B = A^T$ . Entonces:  $U = (2A^T - \mathbb{1})(2A - \mathbb{1})$

$$\Rightarrow = (2SAS - \mathbb{1})(2A - \mathbb{1}) = S(2A - \mathbb{1})S(2A - \mathbb{1}) = [S(2A - \mathbb{1})]^2$$

Donde  $S$  es el operador SWAP,  $S = \sum_{ij} |ji\rangle\langle ij|$

Tomando  $W = (2A - \mathbb{1})$ , el operador de difusión es

$$U = (SW)^2 \tag{7.1}$$

## 7.5. PageRank cuántico

Finalmente, procedemos a cuantizar el algoritmo de PageRank. Partimos del hecho de que el algoritmo de PageRank se puede formular como una caminata aleatoria, cuya matriz de probabilidades es la matriz de Google,  $G$ . Entonces seguimos el procedimiento de Szegedy, sustituyendo  $p_{ij}$  por  $G_{ij}$ .

Ahora, definimos el valor de PageRank cuántico en el paso  $m$  como:

$$I_q(P_i, m) = \left| U^{\dagger m} (\mathbb{1} \otimes |i\rangle\langle i|) \right\rangle$$

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_i |\psi_i\rangle$$

Esto equivale a realizar  $m$  pasos de la caminata con  $|\psi_0\rangle$  como estado inicial y realizar una medida proyectiva sobre  $|i\rangle_2$ .

Nota:  $I_q$  no converge, sino que oscila, así que se toma el centro de las oscilaciones como la medida de importancia de las páginas. Esto se hace promediando  $I_q$  sobre  $m$ :  $\langle I_q(P_i) \rangle = \frac{1}{M} \sum_{m=0}^{M-1} I_q(P_i, m)$

## 7.6. Circuitos de las caminatas cuánticas de Szegedy

Loke y Wang [16] proponen un esquema para construir eficientemente algoritmos de las caminatas cuánticas de Szegedy. Este esquema consiste en separar las reflexiones del algoritmo en distintas etapas y realizar reflexiones alrededor de estados de la base computacional.

Sea el operador de reflexión del operador de difusión

$$W = 2 \sum_i |\psi_i\rangle\langle\psi_i| - \mathbb{1} = 2 \sum_i |i\rangle\langle i| \otimes |\psi'_i\rangle\langle\psi'_i| - \mathbb{1} \quad (7.2)$$

Donde  $|\psi'_i\rangle = \sum_j \sqrt{p_{ji}} |j\rangle_2$

Si le aplicamos la transformación unitaria  $K = \sum_i |i\rangle\langle i| \otimes K_i$  tal que  $U_i |\psi'_i\rangle = |b\rangle$ , donde  $|b\rangle$  es un estado de la base computacional, tendremos:



$$\begin{aligned}
KWK^\dagger &= K(2A - \mathbb{1})K^\dagger = 2KAK^\dagger - \mathbb{1} = 2 \sum_i |i\rangle\langle i| \otimes K_i |\psi'_i\rangle\langle\psi'_i| K_i^\dagger - \mathbb{1} \\
&= 2 \sum_i |i\rangle\langle i| \otimes |b\rangle\langle b| - \mathbb{1} = 2\mathbb{1}_1 \otimes |b\rangle\langle b|_2 - \mathbb{1} = D \quad (7.3)
\end{aligned}$$

Lo cual se puede implementar fácilmente con compuertas de fase controladas, ya que es una reflexión alrededor de un estado de la base computacional del segundo registro. Sin embargo, esto todavía requeriría hallar  $N$   $K_i$  distintos para un grafo de  $N$  nodos. Para disminuir la cantidad de  $K_i$  a hallar, se pueden aprovechar simetrías en la matriz de adyacencia del grafo. Si separamos el grafo en subgrafos cíclicos, bastaría con hallar un  $K_i$  por subgrafo. En los grafos cíclicos, cada fila de la matriz de adyacencia, y de la matriz de Google, es una permutación de la anterior. Lo mismo sucede con los estados asociados a cada uno de los nodos, así que, con un operador de permutación  $T$ , se podrían convertir los estados de todos los nodos de un grafo cíclico en un mismo estado de referencia. Por ejemplo, supongamos un grafo, cuyos estados  $|\psi'_i\rangle$  son:

$$\begin{aligned}
|\psi'_0\rangle &= \begin{pmatrix} \sqrt{0,0375} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \end{pmatrix} & |\psi'_1\rangle &= \begin{pmatrix} \sqrt{0,8875} \\ \sqrt{0,0375} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \end{pmatrix} \\
|\psi'_2\rangle &= \begin{pmatrix} \sqrt{0,8875} \\ \sqrt{0,8875} \\ \sqrt{0,0375} \\ \sqrt{0,8875} \end{pmatrix} & |\psi'_3\rangle &= \begin{pmatrix} \sqrt{0,8875} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \\ \sqrt{0,0375} \end{pmatrix}
\end{aligned} \quad (7.4)$$

Entonces, el operador de permutación

$$T = X(1)\text{CNOT}(1, 0) \quad (7.5)$$

Permite transformar  $|\psi'_1\rangle, |\psi'_2\rangle, |\psi'_3\rangle$  en  $|\psi'_0\rangle$  de la siguiente manera:

$$|\psi'_0\rangle = T^\dagger |\psi'_1\rangle = T^3 |\psi'_1\rangle = T^2 |\psi'_2\rangle = T |\psi'_3\rangle \quad (7.6)$$

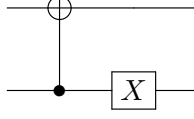


Figura 7.2: Operador de permutación  $T$

Luego, siguiendo esta idea,  $K_i = K_b^\dagger T_i$ , donde  $K_b^\dagger |\psi'_r\rangle = |b\rangle$  y  $T_i |\psi'_i\rangle = |\psi'_r\rangle$ . En otras palabras,  $\{K_b\}$  representa el conjunto de  $K_i$  necesarios después de separar el grafo, uno por cada subgrafo cíclico;  $|\psi'_r\rangle$  es un estado de referencia de cada subgrafo cíclico y podría ser el estado asociado a alguno de los nodos de ese subgrafo;  $T_i$  es un operador que transforma el estado de cada nodo en el estado de referencia del subgrafo cíclico correspondiente. En caso de haber elegido  $|\psi'_r\rangle$  como el estado de alguno de los nodos del subgrafo cíclico, entonces  $T_i$  puede ser un mismo operador de permutación aplicado repetidas veces para todos los nodos del mismo subgrafo, como en el ejemplo anterior.

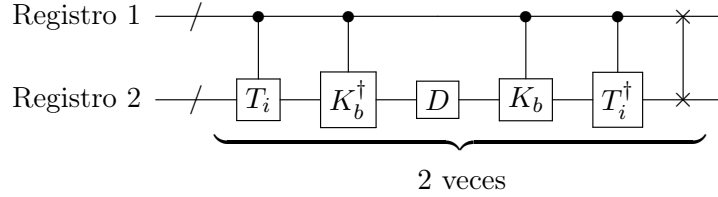


Figura 7.3: Circuito de Loke para las caminatas cuánticas de Szegedy

En resumen, el proceso para construir el circuito de una caminata de Szegedy es:

1. Hallar la matriz de Google del grafo.
2. Separar las filas de la matriz en grupos de filas tales que las filas de cada grupo sean permutaciones una de la otra.
3. Hallar los operadores de permutación  $T_i$ .
4. Hallar los operadores  $K_b$  que conviertan un estado de referencia  $|\phi_{r_i}\rangle$  de cada grupo en un mismo estado de referencia  $|b\rangle$  de la base computacional.
5. Hallar el operador de reflexión  $D$ .

6. Construir el operador de difusión a partir del circuito de la figura 7.3.

Loke y Wang sólo muestran como realizar  $K_i$  para unos pocos casos particulares de caminatas cuánticas de Szegedy que utilizan como ejemplo en su paper. En este trabajo mostramos cómo realizar cualquier  $K_i$  para caminatas cuánticas de Szegedy asociadas a grafos de cuatro nodos.

Todos los coeficientes de los estados involucrados en el algoritmo de PageRank son reales positivos o cero. Es decir, que los qubits individuales que forman estos estados pertenecen todos al arco de semicircunferencia que va de  $+\hat{z}$  a  $-\hat{z}$  pasando por  $+\hat{x}$ . Esto indica que los operadores  $K_i$  deben poder construirse a partir de compuertas  $Ry(\theta)$  y compuertas  $Ry(\theta)$  condicionadas tomando  $0 \leq \theta \leq \pi$ .

En lo que sigue se considerará  $|b\rangle = |0\rangle$  y que los grafos son de cuatro nodos. De esta manera, podemos asumir que

$$K_i = CRy_n(0, 1, \theta_{11})CRy_b(0, 1, \theta_{10})Ry(1, 0, \theta_{00}) \quad (7.7)$$

Entonces, para hallar  $K_i$ , debemos resolver la siguiente ecuación y hallar  $\theta_{11}, \theta_{10}, \theta_{00}$

$$CRy_n(0, 1, \theta_{11})CRy_b(0, 1, \theta_{10})Ry(1, 0, \theta_{00})|0\rangle = |\phi_{r_i}\rangle \quad (7.8)$$

Sin embargo, esto conduce a un sistema de cuatro ecuaciones y tres variables. Para poder resolver este sistema por métodos numéricos se modifica la ecuación de la siguiente manera, sabiendo que  $\theta_x$  debe ser  $n2\pi$ , donde  $n$  es entero.

$$\cos\left(\frac{\theta_{00}}{2}\right)\cos\left(\frac{\theta_{10}}{2}\right) = \sqrt{G_1} \quad (7.9)$$

$$\cos\left(\frac{\theta_{00}}{2}\right)\sin\left(\frac{\theta_{10}}{2}\right) = \sqrt{G_2} \quad (7.10)$$

$$\cos\left(\frac{\theta_{11}}{2}\right)\sin\left(\frac{\theta_{00}}{2}\right) = \sqrt{G_3} \quad (7.11)$$

$$\sin\left(\frac{\theta_{00}}{2}\right)\sin\left(\frac{\theta_{11}}{2}\right) = \sqrt{G_4} \quad (7.12)$$

Esto se puede solucionar recordando la normalización de los estados cuánticos y de las matrices estocásticas, entonces, sabemos que  $G_4 = 1 - (G_1 + G_2 + G_3)$ . Por lo que podemos reducir el sistema de ecuaciones a un de tres variables

$$\cos\left(\frac{\theta_{00}}{2}\right)\cos\left(\frac{\theta_{10}}{2}\right) = \sqrt{G_1} \quad (7.13)$$

$$\cos\left(\frac{\theta_{00}}{2}\right)\sin\left(\frac{\theta_{10}}{2}\right) = \sqrt{G_2} \quad (7.14)$$

$$\cos\left(\frac{\theta_{11}}{2}\right)\sin\left(\frac{\theta_{00}}{2}\right) = \sqrt{G_3} \quad (7.15)$$

Sumando 7.13 y 7.14, y aplicando  $\cos^2(\theta) + \sin^2(\theta) = 1$ , se tiene:

$$\cos^2\left(\frac{\theta_{00}}{2}\right)\cos^2\left(\frac{\theta_{10}}{2}\right) = G_1 \quad (7.16)$$

$$\cos^2\left(\frac{\theta_{00}}{2}\right) = G_1 + G_2 \quad (7.17)$$

$$\cos^2\left(\frac{\theta_{11}}{2}\right)\sin^2\left(\frac{\theta_{00}}{2}\right) = G_3 \quad (7.18)$$

Ahora, sustituyendo 7.17 en 7.16 y 7.18, y volviendo a aplicar la misma propiedad trigonométrica, se tiene:

$$\cos^2\left(\frac{\theta_{10}}{2}\right) = \frac{G_1}{G_1 + G_2} \quad (7.19)$$

$$\cos^2\left(\frac{\theta_{00}}{2}\right) = G_1 + G_2 \quad (7.20)$$

$$\cos^2\left(\frac{\theta_{11}}{2}\right) = \frac{G_3}{1 - (G_1 + G_2)} \quad (7.21)$$

Finalmente, se tiene que los ángulos de las rotaciones deben ser:

$$\theta_{00} = 2 \cos^{-1} \left( \sqrt{G_1 + G_2} \right) \quad (7.22)$$

$$\theta_{10} = 2 \cos^{-1} \left( \sqrt{\frac{G_1}{G_1 + G_2}} \right) \quad (7.23)$$

$$\theta_{11} = 2 \cos^{-1} \left( \sqrt{\frac{G_3}{1 - (G_1 + G_2)}} \right) \quad (7.24)$$

## 7.7. Simulaciones

Se han realizado simulaciones del algoritmo de PageRank con un grafo estrella, un grafo corona, un grafo árbol y un grafo aleatorio, todos de cuatro nodos. Se han realizado simulaciones con y sin pérdidas. El código de las simulaciones se encuentra en el apéndice ??.

### 7.7.1. Grafo estrella

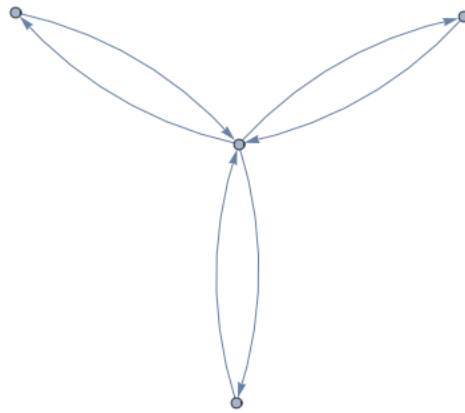


Figura 7.4: Grafo estrella

### 7.7.2. Grafo corona

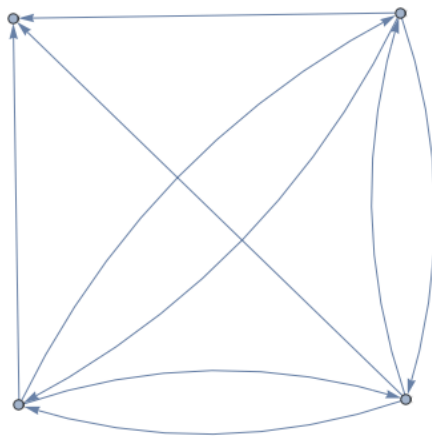


Figura 7.5: Grafo corona

### 7.7.3. Grafo árbol

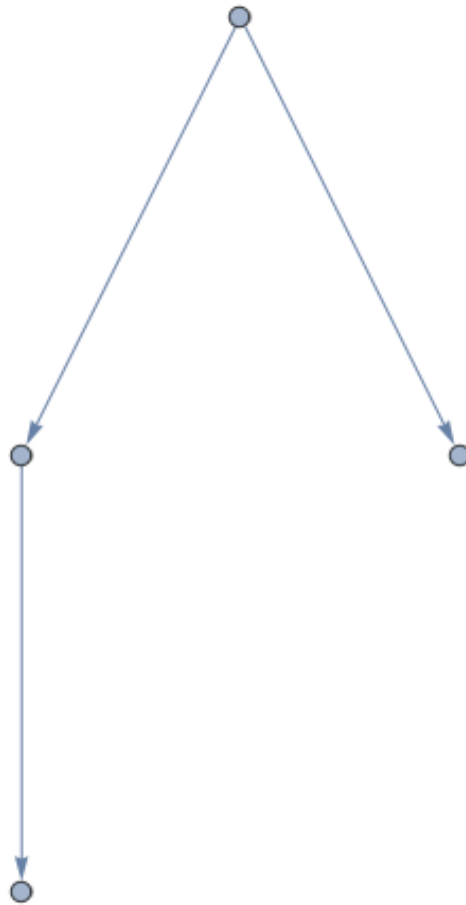


Figura 7.6: Grafo árbol

#### 7.7.4. Grafo aleatorio

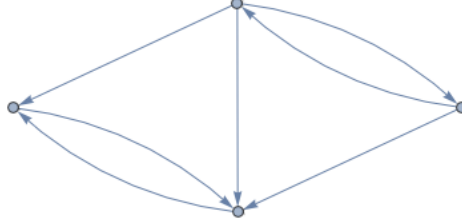


Figura 7.7: Grafo aleatorio

Se ha realizado una simulación del algoritmo de Grover en Wolfram Mathematica implementando  $U_\omega$ ,  $U_s$  y la transformada de Hadamard, directamente, de manera matricial, de acuerdo a las definiciones dadas anteriormente. Por otro lado, se ha realizado una simulación del algoritmo de Grover en Python definiendo todas las operaciones y transformaciones en base a sus construcciones circuitales, a partir de las compuertas nativas de los transmones, resolviendo la ecuación maestra del sistema al aplicar cada compuerta nativa. A la primera la llamaremos la simulación matemática, y a la segunda, simulación circuital. El código de ambas simulaciones se encuentra en el apéndice (APENDICE).

En el caso de la simulación matemática, sólo se ha simulado el caso sin pérdidas. Sin embargo, en el caso de la simulación circuital, se ha simulado el sistema tanto sin pérdidas, como con pérdidas. En el caso del sistema con pérdidas, se ha utilizado la ecuación maestra de Lindblad con los operadores de colapso  $\sigma_{-i}$  y tasa de relajación  $\gamma = 25KHz$ . Primero compararemos las dos simulaciones sin pérdidas para analizar la precisión del solucionador de ecuaciones maestras.

En la figura (FIGURA) se puede observar la gráfica de la evolución de la probabilidad de medir cada estado en cada iteración. Como se puede observar, ambas figuras son bastante similares. Por otro lado, la fidelidad entre los estados finales de ambas simulaciones es  $\langle ++ \rangle$ . Por lo que la precisión del solucionador de ecuaciones maestras causó un error correspondiente a una pérdida de  $\langle ++ \rangle$  de fidelidad.



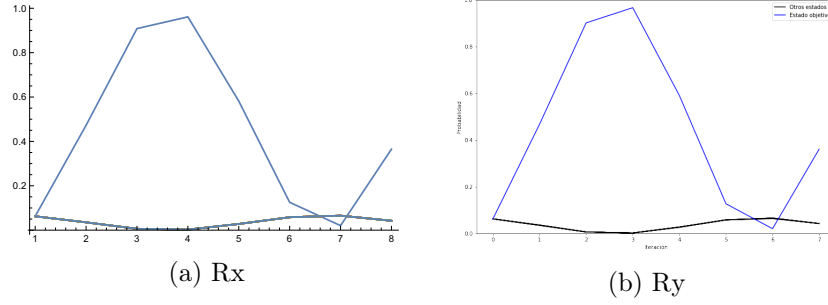


Figura 7.8: Compuertas Rx, Ry y Rz en la esfera de Bloch

Ahora, compararemos los resultados de la simulación circuital con y sin pérdidas. Como se puede ver en la figura (FIGURA), en el caso con pérdidas, los estados que no contienen el valor deseado dejan de tener la misma probabilidad. Los estados que involucran el estado base ganan probabilidad debido a la relajación de los qubits. La fidelidad entre los estados resultantes de los casos con y sin pérdidas es de  $\langle ++ \rangle$ .

Aunque la fidelidad sea tan baja, la probabilidad del estado deseado sigue siendo lo suficientemente alta como para obtener el resultado deseado  $x/x$  veces.

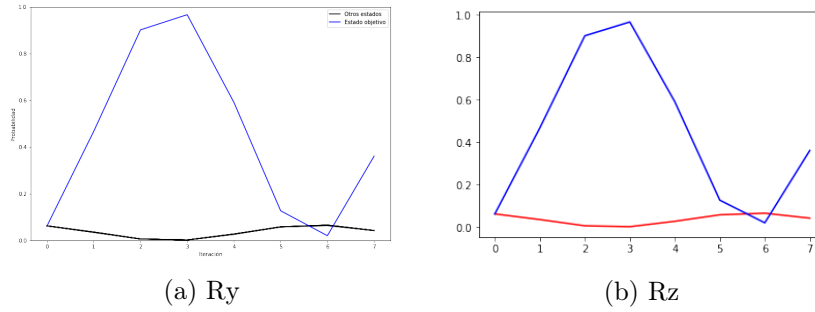


Figura 7.9: Compuertas Rx, Ry y Rz en la esfera de Bloch

# Bibliografía

- [1] Adriano Barenco, Charles H. Bennet, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, Jhon A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 1995.
- [2] Sttiwuer Díaz-Solórzano. Esquemas de medidas. *QIC*, 2014.
- [3] Rudolf Gross and Achim Marx. Applied superconductivity: Josephson effect and superconducting electronics. *Walther-Meißner-Institut*, 2005.
- [4] Onnes H.K. Further experiments with liquid helium. g. on the electrical resistance of pure metals, etc. vi. on the sudden change in the rate at which the resistance of mercury disappears. *Springer, Dordrecht*, 1911.
- [5] A. P. Drozdov, M. I. Erements, I. A. Troyan, V. Ksenofontov, and S. I. Shylin. Conventional superconductivity at 203 kelvin at high pressures in the sulfur hydride system. *Nature*, 525:73–76, 2015.
- [6] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Theory of superconductivity. *Physical Review Journals Archive*, 1957.
- [7] Herbert Fröhlich. Theory of the superconducting state. *Unknown*, 1950.
- [8] M Cyrot. Ginzburg-landau theory for superconductors. *Reports on Progress in Physics*, 36(2):103, 1973.
- [9] Jr. Bascom S. Deaver and William M. Fairbank. Experimental evidence for quantized flux in superconducting cylinders. *Physical Review Letters*, 1961.
- [10] B.D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251 – 253, 1962.

- [11] P. W. Anderson and J. M. Rowell. Probable observation of the josephson superconducting tunneling effect. *Phys. Rev. Lett.*, 10:230–232, Mar 1963.
- [12] Sidney Shapiro. Josephson currents in superconducting tunneling: The effect of microwaves and other observations. *Phys. Rev. Lett.*, 11:80–82, Jul 1963.
- [13] G. Wendin. Quantum information processing with superconducting circuits: a review. *IOP Science*, 2017.
- [14] Alexandre Blais, Jay Gambetta, A. Wallraff, D. I. Schuster, S. M. Girvin, M. H. Devoret, , and R. J. Schoelkopf. Quantum-information processing with circuit quantum electrodynamics. *Physical Review A*, 2007.
- [15] Norbert Schuch and Jens Siewert. Natural two-qubit gate for quantum computation using the xy interaction. *Physical Review A*, 2003.
- [16] T. Loke and J.B. Wang. Efficient quantum circuits for szegedy quantum walks. *Annals of Physics*, 382:64 – 84, 2017.