



Universidad Simón Bolívar
Decanato de Estudios Profesionales
Coordinación de Tecnología e Ingeniería de Electrónica

Diseño y Simulación de Procesadores Cuánticos que Implementen Algoritmos Cuánticos de Búsqueda

Por:
Miguel Casanova
Realizado con la asesoría de:
Enrique Castro y Sttiwuer Diaz

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero Electrónico

Sartenejas, noviembre de 2018

Índice general

Índice de Figuras	7
1. Introducción	10
1.1. Justificación	15
1.2. Objetivos	17
1.2.1. Objetivo General	17
1.2.2. Objetivos Específicos	18
1.3. Fases del Proyecto	18
2. Información cuántica	20
2.1. Operadores lineales	20
2.2. Delta de Kronecker	21
2.3. Operadores hermíticos	21
2.4. Operadores unitarios	22
2.5. Conmutador y anticonmutador	23
2.6. Espacios de Hilbert	24
2.7. Estados cuánticos	26
2.8. Sistemas multipartitos	32
2.9. Postulados de la mecánica cuántica	35
2.10. Entrelazamiento	36
2.11. Qubits	37
2.12. Esfera de Bloch	38
2.13. Matrices de Pauli	39
2.14. Circuitos cuánticos	39
2.15. Compuertas cuánticas de un qubit	42
2.15.1. Compuerta identidad	42
2.15.2. Compuerta X	42
2.15.3. Compuerta Z	43
2.15.4. Compuerta Y	44
2.15.5. Compuerta de Hadamard	44
2.15.6. Compuerta S	45
2.15.7. Compuerta T	45
2.15.8. Compuerta de cambio de fase	46
2.15.9. Compuertas de rotación	46
2.16. Compuertas multiqubit	48

2.16.1. Compuerta CNOT	48
2.16.2. Compuerta SWAP	48
2.16.3. Compuerta $\sqrt{\text{SWAP}}$	49
2.16.4. Compuerta de Ising	50
2.16.5. Compuerta de Toffoli	50
2.16.6. Compuerta de Deutsch	51
2.17. Conjuntos universales de compuertas cuánticas	52
2.18. Criterios de DiVincenzo	52
2.19. Fidelidad	53
2.20. Medidas proyectivas	54
2.21. Sistemas cuánticos abiertos	54
3. Superconductividad	62
3.1. Cuantización macroscópica y superconductividad	62
3.2. La teoría BCS	64
3.3. Cuantización del flujo magnético y efecto tunel Giaver	72
3.4. Efecto Josephson	78
3.5. Componentes de la corriente en las uniones de Josephson	83
3.6. Qubits superconductores	84
3.7. Arquetipos de qubits superconductores	86
3.8. Transmones	86
3.9. Hamiltonianos multiqubit de transmones	88
3.10. Compuertas cuánticas en transmones	90
3.10.1. Rotaciones X-Y	91
3.10.2. Compuerta de entrelazamiento	92
3.10.3. Compuertas compuestas	93
4. El simulador	94
4.1. Parámetros de los sistemas simulados	95
4.2. Compuertas nativas	96
4.2.1. Rx y Ry	96
4.2.2. iSWAP	98
4.3. Compuertas compuestas	99
4.3.1. X	99
4.3.2. Y	99
4.3.3. Rz	99
4.3.4. Z	100
4.3.5. H	100
4.3.6. CNOT	101
4.3.7. SWAP	101
4.3.8. Compuertas condicionales generales	101
4.3.9. CP	105
5. Algoritmo de Grover	113
5.1. El algoritmo	117

5.2. Variaciones y generalizaciones del algoritmo de Grover	118
5.2.1. Algoritmo de amplificación de amplitud	118
5.2.2. Algoritmo de Grover en un paso	121
5.2.3. Optimización del algoritmo de Grover	122
5.3. Simulaciones	123
6. Algoritmo de Shor	128
6.1. Transformada cuántica de Fourier	128
6.2. Estimación de fase	130
6.3. Estimación de orden	133
6.4. Expansión en fracciones continuas	136
6.5. Algoritmo de factorización de Shor	137
6.6. Simulaciones	138
6.6.1. Factorización del número 15	138
6.6.2. Factorización del número 8	141
7. Google PageRank	143
7.1. El algoritmo de remiendo (parcheo) general	146
7.2. Interpretación como una caminata aleatoria	147
7.3. Cuantizando las caminatas aleatorias	148
7.4. Caminata cuántica de Szegedy	148
7.5. PageRank cuántico	150
7.6. Circuitos de las caminatas cuánticas de Szegedy	150
7.7. Simulaciones	155
7.7.1. Grafo estrella	155
7.7.2. Grafo corona	159
7.7.3. Grafo árbol	163
7.7.4. Grafo aleatorio	167
8. Conclusiones	171
A. Cálculos de Hamiltonianos	174
A.1. Régimen rotacional del pulso	174
A.2. Efecto del pulso sobre el qubit	178
A.3. Régimen dispersivo	180
B. Códigos del simulador	185
B.1. Wolfram Mathematica	185
B.2. Python	192
C. Códigos de la simulación del algoritmo de Grover	205
C.1. Wolfram Mathematica	205
C.2. Python	206
D. Códigos de la simulación del algoritmo de Shor	209
D.1. Wolfram Mathematica	209

D.2. Python	212
E. Códigos de la simulación del algoritmo de PageRank	215
E.1. Wolfram Mathematica	215
E.2. Python	218
E.2.1. Grafo estrella	222
E.2.2. Grafo corona	226
E.2.3. Grafo árbol	228
E.2.4. Grafo aleatorio	231

Índice de figuras

2.1. Esfera de Bloch	38
2.2. Compuerta I en la esfera de Bloch	42
2.3. Compuerta X en la esfera de Bloch	43
2.4. Compuerta Z en la esfera de Bloch	43
2.5. Compuerta Y en la esfera de Bloch	44
2.6. Compuerta H en la esfera de Bloch	45
2.7. Compuerta S en la esfera de Bloch	45
2.8. Compuerta T en la esfera de Bloch	46
2.9. Compuerta P en la esfera de Bloch	46
2.10. Compuertas Rx, Ry y Rz en la esfera de Bloch	47
3.1. Diagrama de Feynman de la interacción electrón-fonón-electrón . . .	67
3.2. Construcción geométrica de los posibles electrones candidatos para formar pares de Cooper, siendo $\hbar K$ el momentum del centro de masas.	69
3.3. Cuantización del flujo magnético	74
3.4. Diagrama de energía de una unión metal-aislante-metal en la que no puede haber efecto túnel	76
3.5. Diagrama de energía de una unión metal-aislante-metal en la que puede haber efecto túnel	77
3.6. Diagrama de energía de una unión superconductor-aislante-metal en la que puede haber efecto Giaver	78
3.7. Curva característica I-V de una unión Josephson	82
4.1. Rotaciones en X e Y de 2π	97
4.2. Rotaciones en X e Y de π	97
4.3. Rotaciones en X e Y de $\frac{\pi}{2}$	97
4.4. Compuertas iSWAP y \sqrt{iSWAP} aplicadas a $ 00\rangle$	98
4.5. Compuertas iSWAP y \sqrt{iSWAP} aplicadas a $ 01\rangle$	98
4.6. Compuertas iSWAP y \sqrt{iSWAP} aplicadas a $\frac{ 00\rangle+ 11\rangle}{\sqrt{2}}$	98
4.7. Compuertas iSWAP y \sqrt{iSWAP} aplicadas a $\frac{ 0\rangle+ 1\rangle}{\sqrt{2}} \otimes \frac{ 0\rangle+ 1\rangle}{\sqrt{2}}$	98
5.1. Circuito del algoritmo de Grover, k_{max} desconocido.	115
5.2. Interpretación geométrica del operador difusión	117
5.3. Circuito del algoritmo de Grover.	117
5.4. Evolución de las probabilidades en el algoritmo de Grover sin relajación	124

5.5. Evolución de las probabilidades en el algoritmo de Grover con relajación, $\mathcal{W} = \{0\}$	125
5.6. Evolución de las probabilidades en el algoritmo de amplificación de amplitud sin relajación, $\mathcal{W} = \{9, 13\}$	126
5.7. Evolución de las probabilidades en el algoritmo de amplificación de amplitud sin relajación, $\mathcal{W} = \{4, 5, 12, 13\}$	126
5.8. Evolución de las probabilidades en el algoritmo de amplificación de amplitud con relajación	127
6.1. Distribución de probabilidad en la estimación de fase del algoritmo de Shor sin pérdidas	139
6.2. Distribución de probabilidad en la estimación de fase del algoritmo de Shor sin pérdidas	141
7.1. Grafo aleatorio	143
7.2. Transformación de un grafo al crear la matriz de Google con $\alpha = \frac{1}{2}$	147
7.3. Operador de permutación	152
7.4. Circuito de Loke para las caminatas cuánticas de Szegedy	152
7.5. Circuito de K_i	153
7.6. Grafo estrella	156
7.7. Circuito de K_1 para el grafo estrella	156
7.8. Circuito de K_2 para el grafo estrella	157
7.9. K_b del grafo estrella	157
7.10. T del grafo estrella	157
7.11. Preparación del estado inicial para la caminata en el grafo estrella	157
7.12. Circuito del PageRank cuántico del grafo estrella	158
7.13. PageRank cuántico instantáneo del grafo estrella sin pérdidas	158
7.14. PageRank cuántico promedio del grafo estrella sin pérdidas	158
7.15. PageRank cuántico instantaneo del grafo estrella con y sin pérdidas	159
7.16. PageRank cuántico promedio del grafo estrella con y sin pérdidas	159
7.17. Grafo corona	160
7.18. Circuito de K_1 para el grafo corona	160
7.19. Circuito de K_2 para el grafo corona	161
7.20. K_b del grafo corona	161
7.21. T del grafo corona	161
7.22. Preparación del estado inicial para la caminata en el grafo corona	161
7.23. Circuito del PageRank cuántico del grafo corona	162
7.24. PageRank cuántico instantáneo del grafo corona sin pérdidas	162
7.25. PageRank cuántico promedio del grafo corona sin pérdidas	162
7.26. PageRank cuántico instantaneo del grafo aleatorio con y sin pérdidas	163
7.27. PageRank cuántico promedio del grafo aleatorio con y sin pérdidas	163
7.28. Grafo árbol	163
7.29. Circuito de K_1 para el grafo árbol	164
7.30. Circuito de K_2 para el grafo árbol	164
7.31. Circuito de K_3 para el grafo árbol	165

7.32. K_b del grafo árbol	165
7.33. T del grafo árbol	165
7.34. Preparación del estado inicial para la caminata en el grafo árbol . .	165
7.35. Circuito del PageRank cuántico del grafo árbol	166
7.36. PageRank cuántico instantáneo del grafo árbol sin pérdidas	166
7.37. PageRank cuántico promedio del grafo árbol sin pérdidas	166
7.38. PageRank cuántico instantaneo del grafo árbol con y sin pérdidas .	167
7.39. PageRank cuántico promedio del grafo árbol con y sin pérdidas . .	167
7.40. Grafo aleatorio	167
7.41. Circuito de K_1 para el grafo aleatorio	168
7.42. Circuito de K_2 para el grafo aleatorio	168
7.43. Circuito de K_3 para el grafo aleatorio	169
7.44. K_b del grafo aleatorio	169
7.45. T del grafo aleatorio	169
7.46. Preparación del estado inicial para la caminata en el grafo aleatorio	169
7.47. Circuito del PageRank cuántico del grafo aleatorio	170
7.48. PageRank cuántico instantáneo del grafo aleatorio sin pérdidas . . .	170
7.49. PageRank cuántico promedio del grafo aleatorio sin pérdidas	170
7.50. PageRank cuántico instantaneo del grafo aleatorio con y sin pérdidas	171
7.51. PageRank cuántico promedio del grafo aleatorio con y sin pérdidas .	171

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

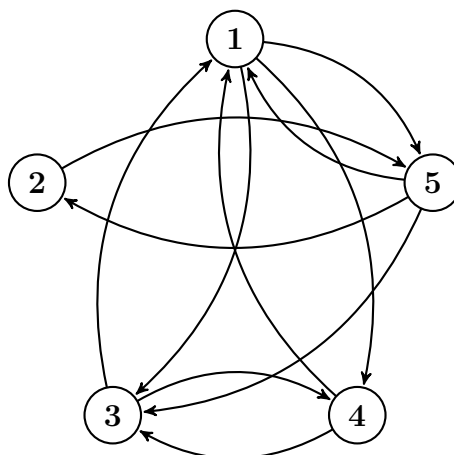


FIGURA 7.1: Grafo de ejemplo para el PageRank clásico

Entonces se deben realizar los siguientes pasos:

1. Determinar la matriz de adyacencia. Algunos autores denotan la matriz 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:

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

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 λ_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 α , 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

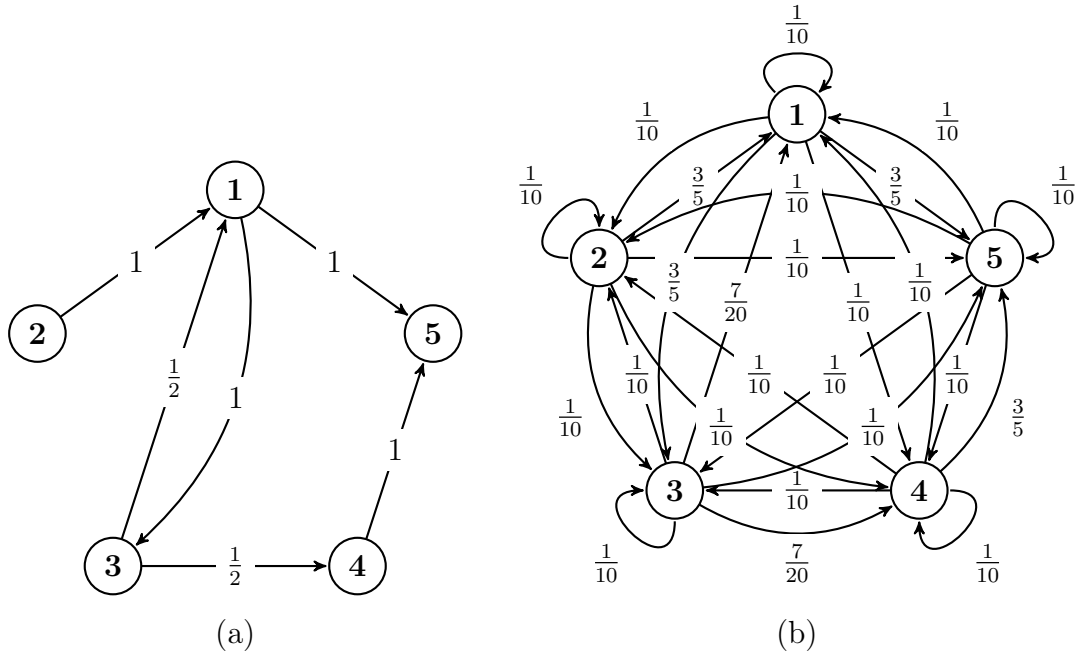


FIGURA 7.2: Transformación de un grafo al crear la matriz de Google con $\alpha = \frac{1}{2}$: Grafo correspondiente a la matriz de adyacencia (a) de la red E (b) remendada de Google G con $\alpha = \frac{1}{2}$

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:

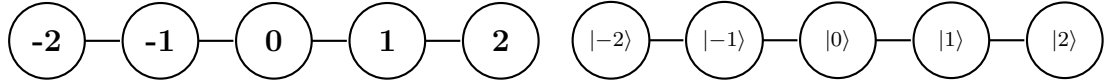
$$\begin{aligned}
 Pr(x^{(n+1)} = p_i) &= \sum_j G_{ij} Pr(x^{(n)} = p_j) \\
 Pr(x^{(n+1)} = p_i) &= \sum_j Pr(x^{(n+1)} = p_i | x^{(n)} = p_j) Pr(x^{(n)} = p_j) \\
 \implies G_{ij} &= Pr(x^{(n+1)} = p_i | x^{(n)} = p_j)
 \end{aligned}$$

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 αE_{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".^{a1} 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.

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

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. También existen caminatas cuánticas continuas, no sólo discretas, pero ellas siguen un esquema distinto de la computación cuántica, donde no se busca que el operador de evolución del sistema corresponda a compuertas cuánticas con las cuales construir el operador de difusión, sino directamente al operador de difusión de la caminata cuántica.

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.

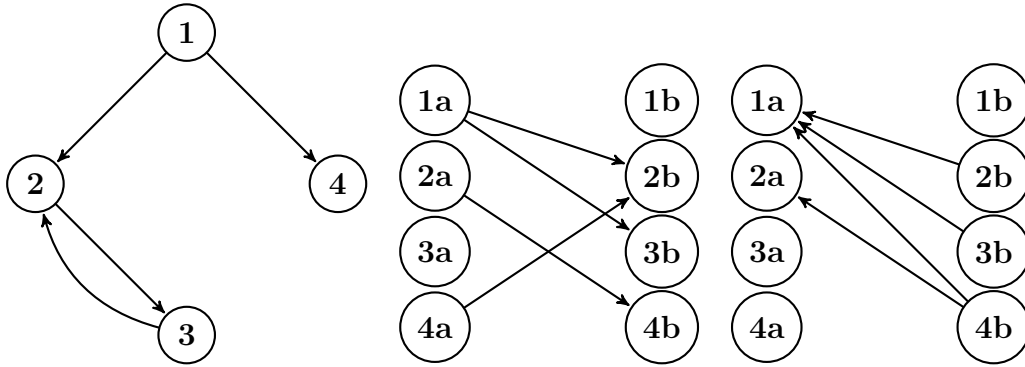
$$|\psi_i\rangle = |i\rangle_1 \otimes \sum_j \sqrt{p_{ji}} |j\rangle_2$$

$$A = \sum_i |\psi_i\rangle\langle\psi_i|$$

$$|\psi_i\rangle = \sum_i \sqrt{p_{ij}} |i\rangle_1 \otimes |i\rangle_2$$

$$B = \sum_j |\phi_j\rangle\langle\phi_j|$$

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:

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

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

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 [70] 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:

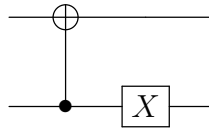
$$\begin{aligned} |\psi'_0\rangle &= \begin{pmatrix} \sqrt{0,0375} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \end{pmatrix} & |\psi'_1\rangle &= \begin{pmatrix} \sqrt{0,8875} \\ \sqrt{0,0375} \\ \sqrt{0,8875} \\ \sqrt{0,8875} \end{pmatrix} \\ |\psi'_2\rangle &= \begin{pmatrix} \sqrt{0,8875} \\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

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 7.3: 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.

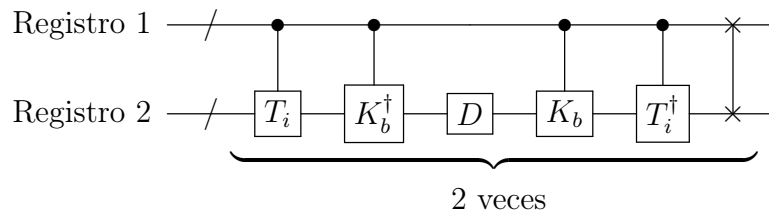


FIGURA 7.4: Circuito de Loke para las caminatas cuánticas de Szegedy

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 7.4.

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

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

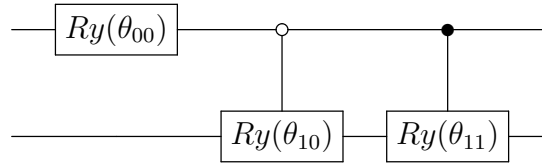


FIGURA 7.5: Circuito de K_i

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

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

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 θ_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 [E](#).

7.7.1. Grafo estrella

En la figura [7.6](#) se observa el grafo estrella utilizado para esta simulación. Este grafo es simétrico, por lo tanto, su matriz de adyacencia es hermítica. Este no es el caso, sin embargo, una vez convertimos el grafo en uno ponderado. La matriz estocástica E no es hermítica, y por extensión la matriz de Google tampoco lo es.

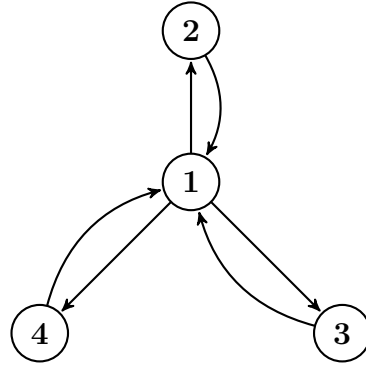


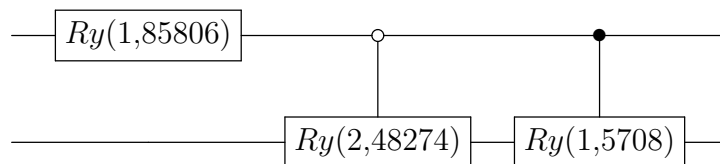
FIGURA 7.6: Grafo estrella

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (7.25)$$

$$E = \begin{pmatrix} 0 & 1 & 1 & 1 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \end{pmatrix} \quad (7.26)$$

$$G = \begin{pmatrix} \frac{3}{80} & \frac{71}{80} & \frac{71}{80} & \frac{71}{80} \\ \frac{77}{240} & \frac{3}{80} & \frac{3}{80} & \frac{3}{80} \\ \frac{77}{240} & \frac{3}{80} & \frac{3}{80} & \frac{3}{80} \\ \frac{77}{240} & \frac{3}{80} & \frac{3}{80} & \frac{3}{80} \end{pmatrix} \quad (7.27)$$

Los tres nodos exteriores de este grafo, además, forman lo que hemos llamado un subgrafo cíclico, por lo que para este grafo necesitaremos dos operadores K_i , uno para estos tres nodos y otro para el nodo central. Las figuras 7.7 - 7.12 muestran cómo construir el circuito del operador de difusión de la caminata de Szegedy asociada a la matriz de Google de este grafo.

FIGURA 7.7: Circuito de K_1 para el grafo estrella

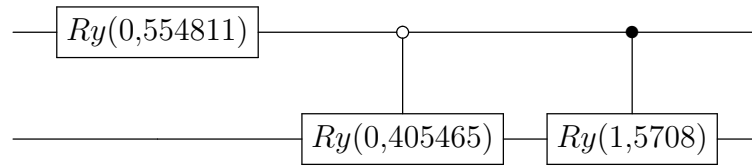


FIGURA 7.8: Circuito de K_2 para el grafo estrella

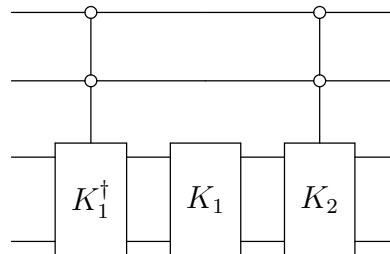


FIGURA 7.9: K_b del grafo estrella

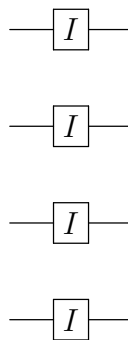


FIGURA 7.10: T del grafo estrella

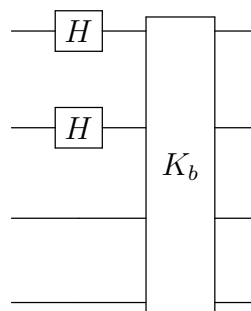


FIGURA 7.11: Preparación del estado inicial para la caminata en el grafo estrella

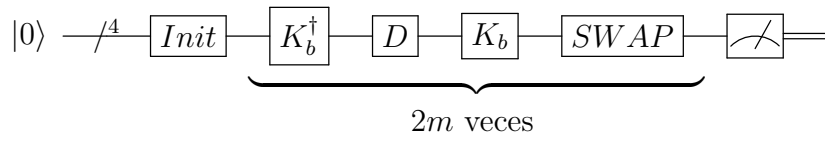
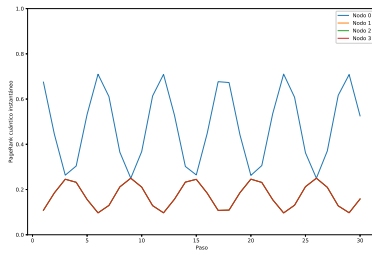
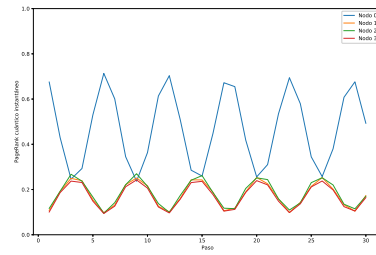


FIGURA 7.12: Circuito del PageRank cuántico del grafo estrella

En la figura 7.13 se puede observar la gráfica del PageRank cuántico instantáneo sin relajación. En esta figura se puede observar claramente la naturaleza sinusoidal de este algoritmo, por estar basado en reflexiones. Como se puede observar, ambas figuras son bastante similares. La diferencia está en que en el caso de la simulación matemática, el PageRank cuántico de los tres nodos exteriores se solapan perfectamente, mientras que en la simulación circuital, no. Esto se debe a la limitada precisión numérica del solucionador de ecuaciones maestras.



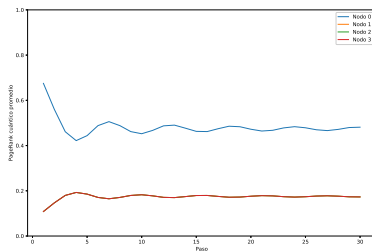
(A) Wolfram Mathematica



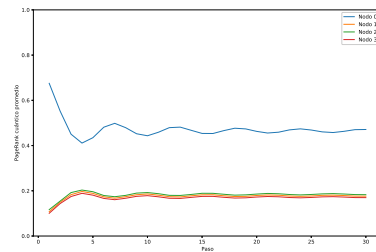
(B) Python

FIGURA 7.13: PageRank cuántico instantáneo del grafo estrella sin pérdidas

En el caso del PageRank promedio, en la figura 7.14 podemos ver cómo este valor sí converge, a diferencia del PageRank instantaneo, el cuál oscila alrededor del valor de convergencia del PageRank promedio.



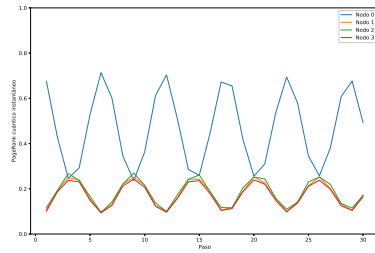
(A) Wolfram Mathematica



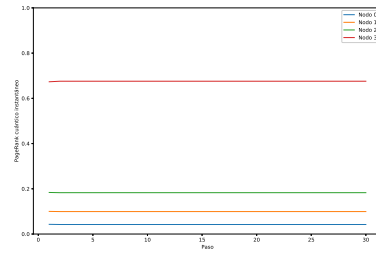
(B) Python

FIGURA 7.14: PageRank cuántico promedio del grafo estrella sin pérdidas

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, $\langle ++ \rangle$

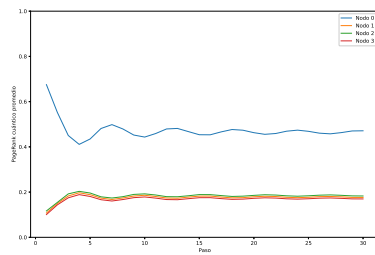


(A) Sin relajación

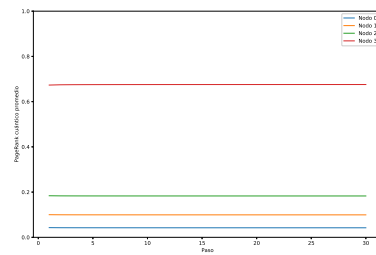


(B) Con relajación

FIGURA 7.15: PageRank cuántico instantáneo del grafo estrella con y sin pérdidas



(A) Sin relajación



(B) Con relajación

FIGURA 7.16: PageRank cuántico promedio del grafo estrella con y sin pérdidas

7.7.2. Grafo corona

En la figura 7.17 se observa el grafo corona utilizado para esta simulación. Este grafo no es simétrico, por lo tanto, ni su matriz de adyacencia A , ni su matriz estocástica E , ni su matriz de Google G son hermiticas.

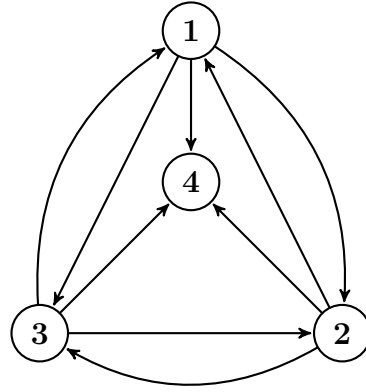


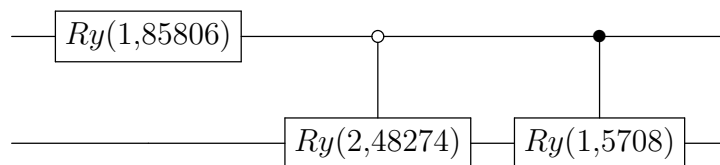
FIGURA 7.17: Grafo corona

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad (7.28)$$

$$E = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{4} \end{pmatrix} \quad (7.29)$$

$$G = \begin{pmatrix} \frac{3}{80} & \frac{77}{240} & \frac{77}{240} & \frac{1}{4} \\ \frac{77}{240} & \frac{3}{80} & \frac{77}{240} & \frac{1}{4} \\ \frac{77}{240} & \frac{77}{240} & \frac{3}{80} & \frac{1}{4} \\ \frac{77}{240} & \frac{77}{240} & \frac{77}{240} & \frac{1}{4} \end{pmatrix} \quad (7.30)$$

Como en el grafo corona, en este grafo, los tres nodos exteriores un subgrafo cíclico, por lo que para este grafo también necesitaremos dos operadores K_i , uno para estos tres nodos y otro para el nodo central. Las figuras 7.18 - 7.23 muestran cómo construir el circuito del operador de difusión de la caminata de Szegedy asociada a la matriz de Google de este grafo.

FIGURA 7.18: Circuito de K_1 para el grafo corona

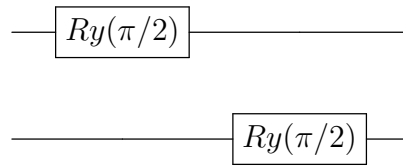


FIGURA 7.19: Circuito de K_2 para el grafo corona

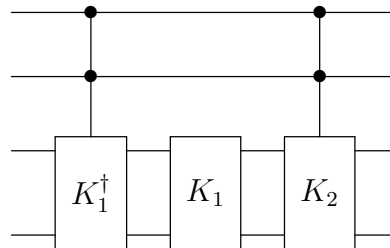


FIGURA 7.20: K_b del grafo corona

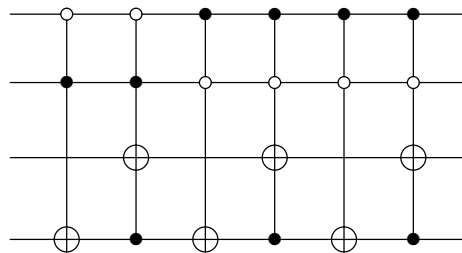


FIGURA 7.21: T del grafo corona

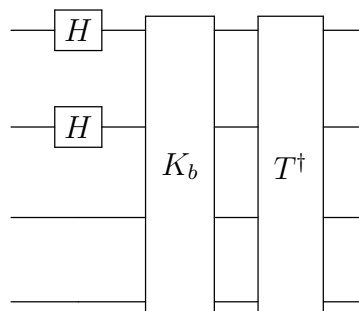


FIGURA 7.22: Preparación del estado inicial para la caminata en el grafo corona

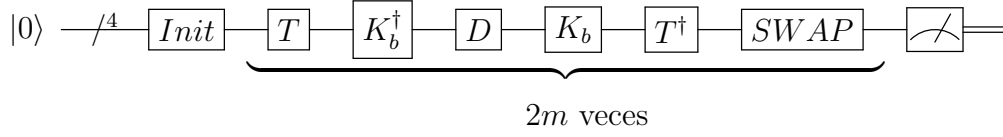
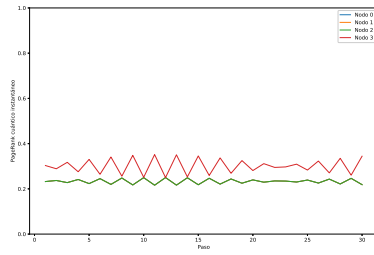
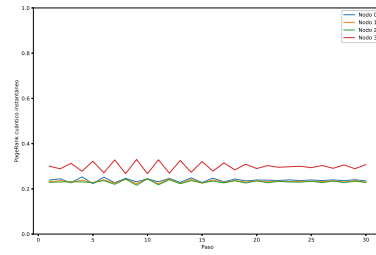


FIGURA 7.23: Circuito del PageRank cuántico del grafo corona

En la figura 7.24 se pueden observar las gráficas del PageRank cuántico instantáneo sin relajación, resultantes de la simulación matemática y de la simulación circuital. Como se puede observar, ambas figuras son bastante similares. En el caso de la simulación matemática, el PageRank cuántico de los tres nodos exteriores se solapan perfectamente, mientras que en la simulación circuital, no. Otra diferencia está en que las oscilaciones del PageRank cuántico en la simulación matemática son más amplias que las de la simulación circuital. Esto se debe a la limitada precisión numérica del solucionador de ecuaciones maestras.

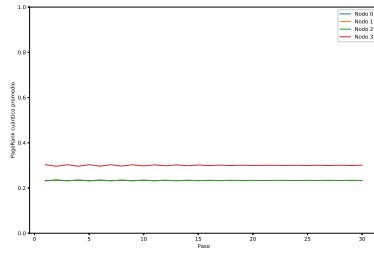


(A) Wolfram Mathematica

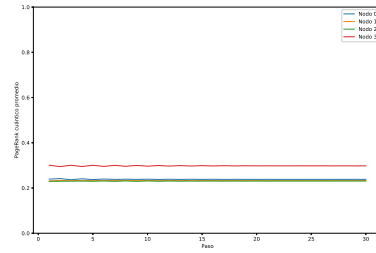


(B) Python

FIGURA 7.24: PageRank cuántico instantáneo del grafo corona sin pérdidas

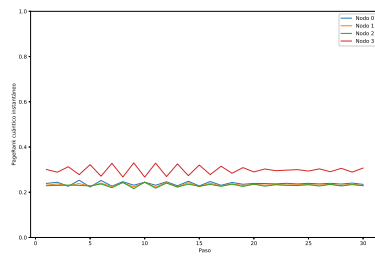


(A) Wolfram Mathematica

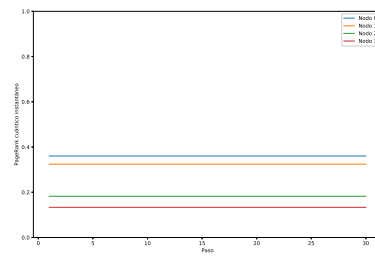


(B) Python

FIGURA 7.25: PageRank cuántico promedio del grafo corona sin pérdidas

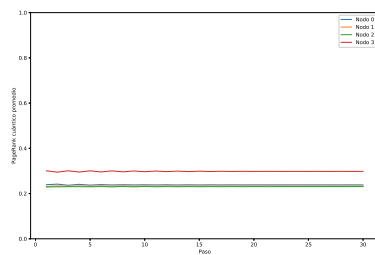


(A) Sin relajación

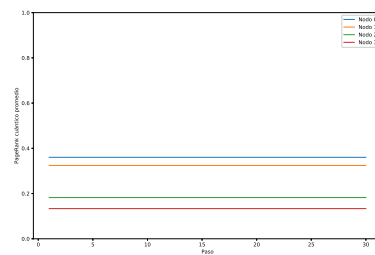


(B) Con relajación

FIGURA 7.26: PageRank cuántico instantáneo del grafo aleatorio con y sin pérdidas



(A) Sin relajación



(B) Con relajación

FIGURA 7.27: PageRank cuántico promedio del grafo aleatorio con y sin pérdidas

7.7.3. Grafo árbol

En la figura 7.28 se observa el grafo árbol utilizado para esta simulación. Este grafo no es simétrico, por lo tanto, ni su matriz de adyacencia A , ni su matriz estocástica E , ni su matriz de Google G son hermiticas.

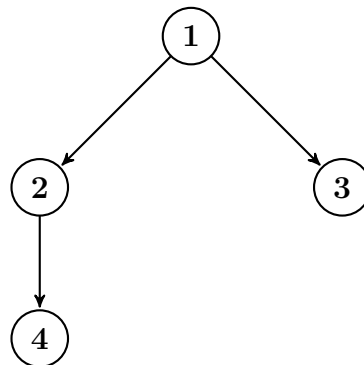


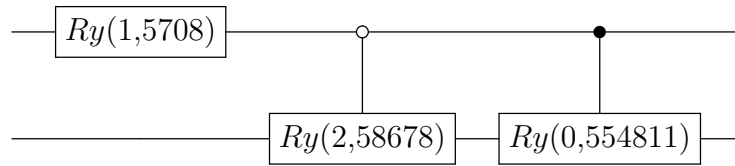
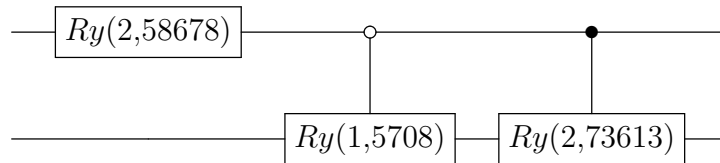
FIGURA 7.28: Grafo árbol

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (7.31)$$

$$E = \begin{pmatrix} 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \quad (7.32)$$

$$G = \begin{pmatrix} \frac{3}{80} & \frac{3}{80} & \frac{1}{4} & \frac{1}{4} \\ \frac{37}{80} & \frac{3}{80} & \frac{1}{4} & \frac{1}{4} \\ \frac{37}{80} & \frac{3}{80} & \frac{1}{4} & \frac{1}{4} \\ \frac{3}{80} & \frac{71}{80} & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \quad (7.33)$$

En este grafo, los dos nodos hojas, 3 y 4, forman un subgrafo cíclico. El nodo raíz, 1, es su propio grafo cíclico, igual que el nodo 2. Por lo tanto, para este grafo necesitaremos tres operadores K_i , uno para los nodos 3 y 4, otro para el nodo 1 y otro para el nodo 2. Las figuras 7.29 - 7.35 muestran cómo construir el circuito del operador de difusión de la caminata de Szegedy asociada a la matriz de Google de este grafo.

FIGURA 7.29: Circuito de K_1 para el grafo árbolFIGURA 7.30: Circuito de K_2 para el grafo árbol

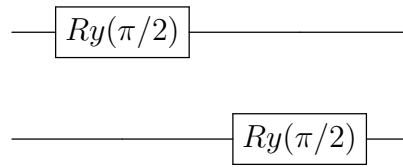


FIGURA 7.31: Circuito de K_3 para el grafo árbol

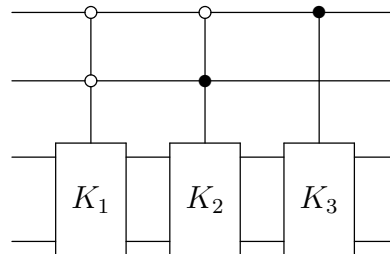


FIGURA 7.32: K_b del grafo árbol

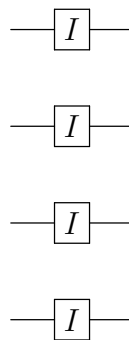


FIGURA 7.33: T del grafo árbol

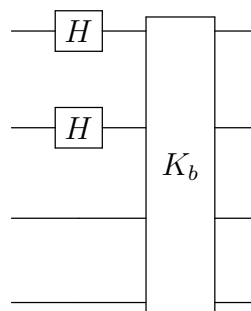


FIGURA 7.34: Preparación del estado inicial para la caminata en el grafo árbol

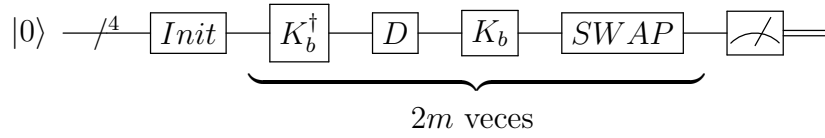
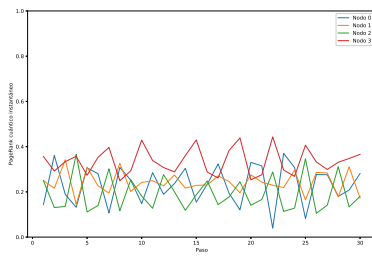
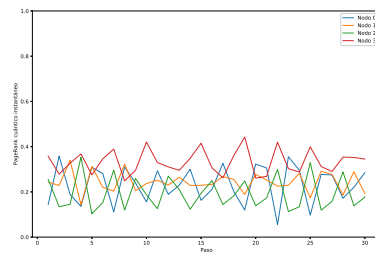


FIGURA 7.35: Circuito del PageRank cuántico del grafo árbol

En la figura 7.36 se pueden observar las gráficas del PageRank cuántico instantáneo sin relajación, resultantes de la simulación matemática y de la simulación circuital. Como se puede observar, ambas figuras son bastante similares. De la misma manera que con el grafo corona, las oscilaciones en la simulación circuital son de menor amplitud.

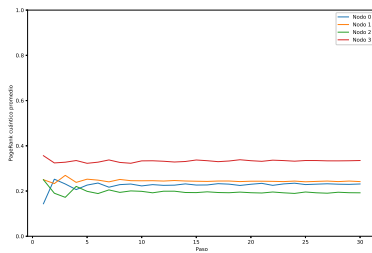


(A) Wolfram Mathematica

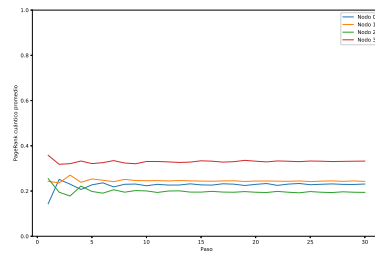


(B) Python

FIGURA 7.36: PageRank cuántico instantáneo del grafo árbol sin pérdidas



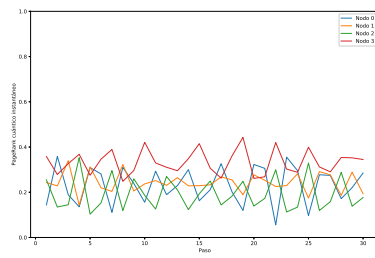
(A) Wolfram Mathematica



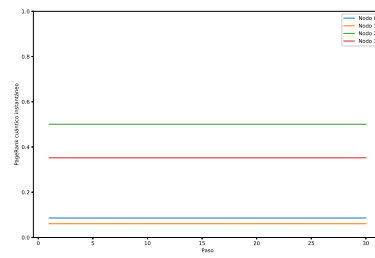
(B) Python

FIGURA 7.37: PageRank cuántico promedio del grafo árbol sin pérdidas

En el caso con pérdidas, presentado en la figura 7.38, se observa que el PageRank cuántico es constante. Esto se debe a lo larga que es cada iteración. En el tiempo que realiza una iteración, los qubits ya le han entregado al entorno la energía introducida por las primeras compuertas, así que el estado final de cada iteración se aproxima a la aplicación de sólo las últimas compuertas de la iteración al estado base.

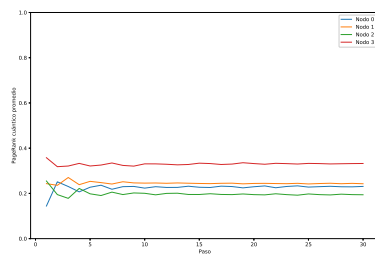


(A) Sin relajación

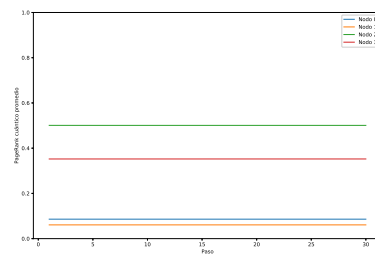


(B) Con relajación

FIGURA 7.38: PageRank cuántico instantáneo del grafo árbol con y sin pérdidas



(A) Sin relajación



(B) Con relajación

FIGURA 7.39: PageRank cuántico promedio del grafo árbol con y sin pérdidas

7.7.4. Grafo aleatorio

En la figura 7.28 se observa último grafo utilizado para las simulaciones y fue construido de manera aleatoria. Este grafo no es simétrico, por lo tanto, ni su matriz de adyacencia A , ni su matriz estocástica E , ni su matriz de Google G son hermíticas.

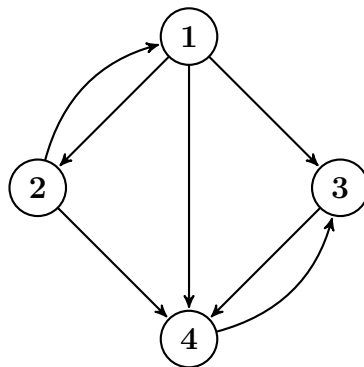


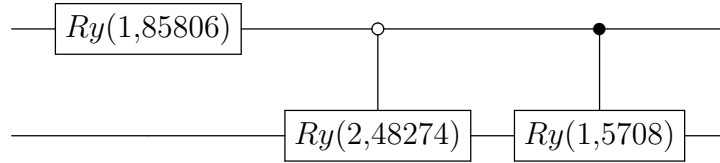
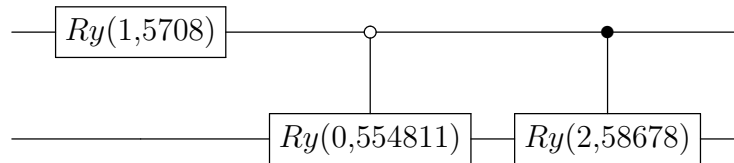
FIGURA 7.40: Grafo aleatorio

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad (7.34)$$

$$E = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{2} & 1 & 0 \end{pmatrix} \quad (7.35)$$

$$G = \begin{pmatrix} \frac{3}{80} & \frac{37}{80} & \frac{3}{80} & \frac{3}{80} \\ \frac{77}{240} & \frac{3}{80} & \frac{3}{80} & \frac{3}{80} \\ \frac{77}{240} & \frac{3}{80} & \frac{3}{80} & \frac{71}{80} \\ \frac{77}{240} & \frac{37}{80} & \frac{71}{80} & \frac{3}{80} \end{pmatrix} \quad (7.36)$$

En este grafo, al igual que en el caso anterior, los nodos 3 y 4 forman un subgrafo cíclico, y los nodos 1 y 2 son sus propios subgrafos cíclicos. Por lo tanto, para este grafo también necesitaremos tres operadores K_i , uno para los nodos 3 y 4, otro para el nodo 1 y otro para el nodo 2. Las figuras 7.41 - 7.47 muestran cómo construir el circuito del operador de difusión de la caminata de Szegedy asociada a la matriz de Google de este grafo.

FIGURA 7.41: Circuito de K_1 para el grafo aleatorioFIGURA 7.42: Circuito de K_2 para el grafo aleatorio

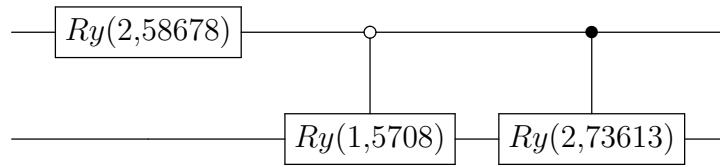


FIGURA 7.43: Circuito de K_3 para el grafo aleatorio

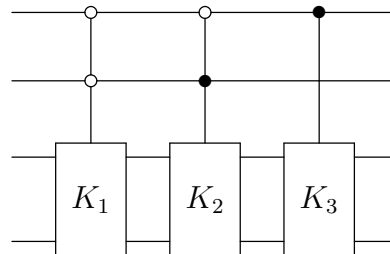


FIGURA 7.44: K_b del grafo aleatorio

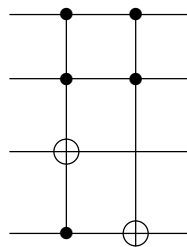


FIGURA 7.45: T del grafo aleatorio

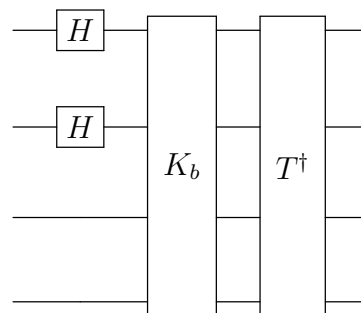


FIGURA 7.46: Preparación del estado inicial para la caminata en el grafo aleatorio

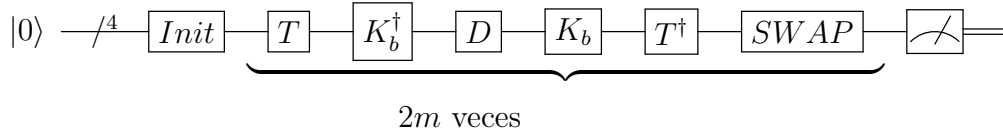
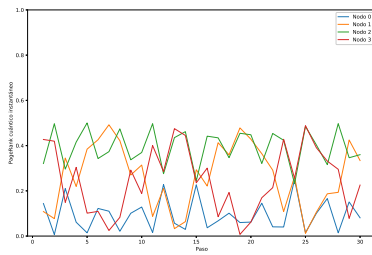
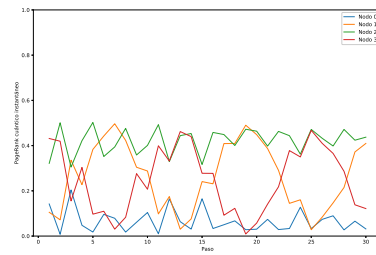


FIGURA 7.47: Circuito del PageRank cuántico del grafo aleatorio

En la figura 7.48 se pueden observar las gráficas del PageRank cuántico instantáneo sin relajación, resultantes de la simulación matemática y de la simulación circuital. Como se puede observar, ambas figuras son bastante similares. De la misma manera que con el grafo corona, las oscilaciones en la simulación circuital son de menor amplitud. También se nota que las curvas son más suaves en el caso circuital. Por ejemplo, en la curva del nodo 3, la no monotonicidad en los tramos de subida o bajada es menor.

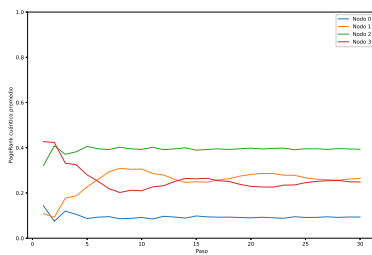


(A) Wolfram Mathematica

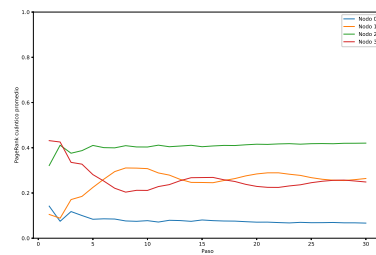


(B) Python

FIGURA 7.48: PageRank cuántico instantáneo del grafo aleatorio sin pérdidas



(A) Wolfram Mathematica

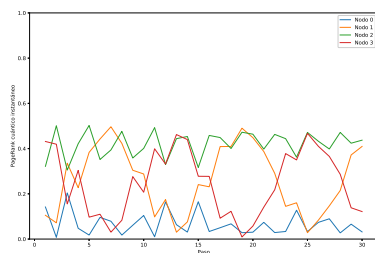


(B) Python

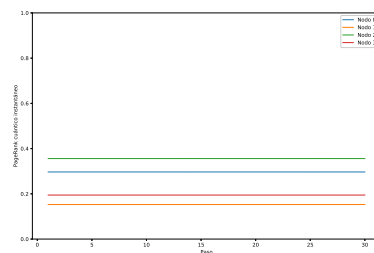
FIGURA 7.49: PageRank cuántico promedio del grafo aleatorio sin pérdidas

En el caso con pérdidas, presentado en la figura 7.50, se observa que el PageRank cuántico es constante. Esto se debe a lo larga que es cada iteración. En el tiempo que realiza una iteración, los qubits ya le han entregado al entorno la energía introducida por las primeras compuertas, así que el estado final de cada iteración

se aproxima a la aplicación de sólo las últimas compuertas de la iteración al estado base.

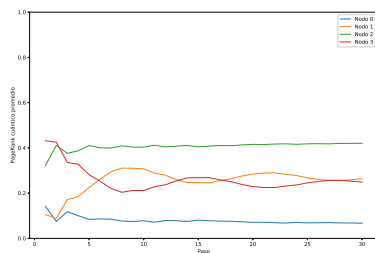


(A) Sin relajación

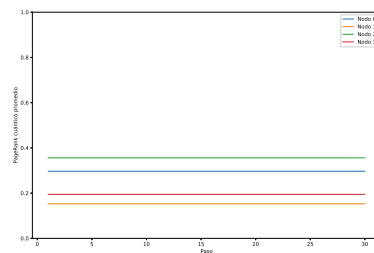


(B) Con relajación

FIGURA 7.50: PageRank cuántico instantáneo del grafo aleatorio con y sin pérdidas



(A) Sin relajación



(B) Con relajación

FIGURA 7.51: PageRank cuántico promedio del grafo aleatorio con y sin pérdidas

Apéndice

Bibliografía

- [1] Masahito Hayashi, Satoshi Ishizaka, Akinori Kawachi, Gen Kimura, and Tomohiro Ogawa. Introduction to quantum information science. *Graduate Texts in Physics*, 2015.
- [2] Jonathan A. Jones and Dieter Jaksch. Quantum information, computation and communication. 2009.
- [3] Michael A. Nielsen and Isaac L. Chuang. Quantum computation and quantum information. 2009.
- [4] Mikio Nakahara. *Quantum computing : from linear algebra to physical realizations*. CRC Press, Boca Raton, 2008.
- [5] G Wendin. Quantum information processing with superconducting circuits: a review. *Reports on Progress in Physics*, 80(10):106001, 2017.
- [6] M. H. Devoret and R. J. Schoelkopf. Superconducting circuits for quantum information: An outlook. *Science*, 339(6124):1169–1174, Mar 2013.
- [7] Yong Hu, Y. X. Zhao, Zheng-Yuan Xue, and Z. D. Wang. Realizing universal quantum gates with topological bases in quantum-simulated superconducting chains. *npj Quantum Information*, 3(1), Mar 2017.
- [8] Davide Rotta, Fabio Sebastiano, Edoardo Charbon, and Enrico Prati. Quantum information density scaling and qubit operation time constraints of cmos silicon-based quantum computer architectures. *npj Quantum Information*, 3(1), Jun 2017.
- [9] Guilherme Tosi, Fahd A. Mohiyaddin, Vivien Schmitt, Stefanie Tenberg, Rajib Rahman, Gerhard Klimeck, and Andrea Morello. Silicon quantum processor with robust long-distance qubit couplings. *Nature Communications*, 8(1), Sep 2017.

- [10] Nicholas C. Harris, Darius Bunandar, Mihir Pant, Greg R. Steinbrecher, Jacob Mower, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, and Dirk Englund. Large-scale quantum photonic circuits in silicon. *Nanophotonics*, 5(3), Jan 2016.
- [11] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 76(4), Oct 2007.
- [12] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, 1996.
- [13] Andreas Dewes, Romain Lauro, Florian R. Ong, Vivient Schmitt, Perola Milman, Patrice Bertet, Denis Vion, and Daniel Esteve. Quantum speeding-up of computation demonstrated in a superconducting two-qubit processor. *Physical Review B*, 85(14), Apr 2012.
- [14] Taouk Said, Abdelhaq Chouikh, K. Essammouni, and Mohamed Bennai. Implementation of grover quantum search algorithm with two transmon qubits via circuit qed. *Quantum Physics Letters*, 6(1):29–35, Apr 2017.
- [15] L. DiCarlo, J. M. Chow, J. M. Gambetta, Lev S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and et al. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature*, 460(7252):240–244, Jun 2009.
- [16] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, Dec 1992.
- [17] Jie Zhao, Xinsheng Tan, Dong Lan, Haiteng Wu, Guangming Xue, Haifeng Yu, and Yang Yu. Implementation of refined deutsch-jozsa algorithm in a superconducting qutrit system. *physica status solidi (b)*, 254(5):1600640, Nov 2016.
- [18] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, Jan 1999.

- [19] Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen. Building logical qubits in a superconducting quantum computing system. *npj Quantum Information*, 3(1), Jan 2017.
- [20] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt. Realization of a scalable shor algorithm. *Science*, 351(6277):1068–1070, Mar 2016.
- [21] Indranil Chakrabarty, Shahzor Khan, and Vanshdeep Singh. Dynamic grover search: applications in recommendation systems and optimization problems. *Quantum Information Processing*, 16(6), Apr 2017.
- [22] K V Gubaidullina and S A Chivilikhin. Theoretical research of the distortion of quantum circuit in grover’s algorithm. *Journal of Physics: Conference Series*, 735(1):012074, 2016.
- [23] Bin Ye, Tingzhong Zhang, Liang Qiu, and Xuesong Wang. Quantum discord and entanglement in grover search algorithm. *Open Physics*, 14(1), Jan 2016.
- [24] Raouf Dridi and Hedayat Alghassi. Prime factorization using quantum annealing and computational algebraic geometry. *Scientific Reports*, 7(1), Feb 2017.
- [25] Niklas Johansson and Jan Åke Larsson. Realization of shor’s algorithm at room temperature, 2017.
- [26] Y. H. Lee, M. Khalil-Hani, and M. N. Marsono. An fpga-based quantum computing emulation framework based on serial-parallel architecture. *International Journal of Reconfigurable Computing*, 2016:1–18, 2016.
- [27] T. H. Johnson, J. D. Biamonte, S. R. Clark, and D. Jaksch. Solving search problems by strongly simulating quantum circuits. *Scientific Reports*, 3(1), Feb 2013.
- [28] Alexandru Paler, Ilia Polian, Kae Nemoto, and Simon J Devitt. Fault-tolerant, high-level quantum circuits: form, compilation and description. *Quantum Science and Technology*, 2(2):025003, 2017.
- [29] Yudong Cao, Anmer Daskin, Steven Frankel, and Sabre Kais. Quantum circuit design for solving linear systems of equations. *Molecular Physics*, 110(15-16):1675–1680, Aug 2012.

- [30] Yarui Zheng, Chao Song, Ming-Cheng Chen, Benxiang Xia, Wuxin Liu, Qiujiang Guo, Libo Zhang, Da Xu, Hui Deng, Keqiang Huang, and et al. Solving systems of linear equations with a superconducting quantum processor. *Physical Review Letters*, 118(21), May 2017.
- [31] G. D. Paparo and M. A. Martin-Delgado. Google in a quantum network. *Scientific Reports*, 2(1), Jun 2012.
- [32] G. D. Paparo, M. Müller, F. Comellas, and M. A. Martin-Delgado. Quantum google algorithm. *The European Physical Journal Plus*, 129(7), Jul 2014.
- [33] Josh A. Izaac, Xiang Zhan, Zhihao Bian, Kunkun Wang, Jian Li, Jingbo B. Wang, and Peng Xue. Centrality measure based on continuous-time quantum walks and experimental realization. *Physical Review A*, 95(3), Mar 2017.
- [34] Quantum manifesto.
- [35] Christoph Simon. Towards a global quantum network. *Nature Photonics*, 11(11):678–680, Oct 2017.
- [36] Bikash K. Behera, Anindita Banerjee, and Prasanta K. Panigrahi. Experimental realization of quantum cheque using a five-qubit quantum computer. *Quantum Information Processing*, 16(12), Nov 2017.
- [37] Fei Yan, Abdullah M. Iliyasu, and Phuc Q. Le. Quantum image processing: A review of advances in its security technologies. *International Journal of Quantum Information*, 15(03):1730001, Apr 2017.
- [38] Yu-Bo Sheng and Lan Zhou. Distributed secure quantum machine learning. *Science Bulletin*, 62(14):1025–1029, Jul 2017.
- [39] Jingyuan Linda Zhang, Konstantinos G. Lagoudakis, Yan-Kai Tzeng, Constantin Dory, Marina Radulaski, Yousif Kelaita, Kevin A. Fischer, Shuo Sun, Zhi-Xun Shen, Nicholas A. Melosh, and et al. Complete coherent control of silicon vacancies in diamond nanopillars containing single defect centers. *Optica*, 4(11):1317, Oct 2017.
- [40] Dong-Ling Deng, Xiaopeng Li, and S. Das Sarma. Quantum entanglement in neural network states. *Physical Review X*, 7(2), May 2017.
- [41] Jialin Chen, Lingli Wang, and Edoardo Charbon. A quantum-implementable neural network model. *Quantum Information Processing*, 16(10), Aug 2017.
- [42] Alberto Galindo and Pedro Pascual. Quantum mechanics i. 1990.

- [43] Yan Wang, YaoMin Di, and HaiRui Wei. Controllability of spin 1 systems and realization of ternary swap gate in two spin 1 systems coupled with ising interaction. *Science China Physics, Mechanics and Astronomy*, 53(10):1873–1877, Aug 2010.
- [44] G.F. Mkrtchian. Universal quantum logic gates in a scalable ising spin quantum computer. *Physics Letters A*, 372(32):5270–5273, Aug 2008.
- [45] 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.
- [46] Sttiwuer Díaz-Solórzano. Esquemas de medidas. *QIC*, 2014.
- [47] A. Kossakowski. On quantum statistical mechanics of non-hamiltonian systems. *Reports on Mathematical Physics*, 3(4):247–274, Dec 1972.
- [48] G. Lindblad. On the generators of quantum dynamical semigroups. *Communications in Mathematical Physics*, 48(2):119–130, Jun 1976.
- [49] Rudolf Gross and Achim Marx. Applied superconductivity: Josephson effect and superconducting electronics. *Walther-Meißner-Institut*, 2005.
- [50] 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.
- [51] 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.
- [52] Michael Tinkham. *Introduction to superconductivity*. McGraw Hill, New York, 1996.
- [53] J. Bardeen, L. N. Cooper, and J. R. Schrieffer. Theory of superconductivity. *Physical Review Journals Archive*, 1957.
- [54] Herbert Fröhlich. Theory of the superconducting state. *Unknown*, 1950.
- [55] M Cyrot. Ginzburg-landau theory for superconductors. *Reports on Progress in Physics*, 36(2):103, 1973.

- [56] Jr. Bascom S. Deaver and William M. Fairbank. Experimental evidence for quantized flux in superconducting cylinders. *Physical Review Letters*, 1961.
- [57] R Doll and M Naebauer. Experimental proof of magnetic flux quantization in a superconducting ring. *Physical Review Letters - PHYS REV LETT*, 7:51–52, 07 1961.
- [58] Ivar Giaever. Electron tunneling between two superconductors. *Phys. Rev. Lett.*, 5:464–466, Nov 1960.
- [59] B.D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251 – 253, 1962.
- [60] P. W. Anderson and J. M. Rowell. Probable observation of the josephson superconducting tunneling effect. *Phys. Rev. Lett.*, 10:230–232, Mar 1963.
- [61] Sidney Shapiro. Josephson currents in superconducting tunneling: The effect of microwaves and other observations. *Phys. Rev. Lett.*, 11:80–82, Jul 1963.
- [62] G. Wendin. Quantum information processing with superconducting circuits: a review. *IOP Science*, 2017.
- [63] Charles A. Schmuttenmaer. Exploring dynamics in the far-infrared with terahertz spectroscopy. *Chemical Reviews*, 104(4):1759–1780, 2004. PMID: 15080711.
- [64] 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.
- [65] Norbert Schuch and Jens Siewert. Natural two-qubit gate for quantum computation using the xy interaction. *Physical Review A*, 2003.
- [66] G. Brassard and P. Hoyer. An exact quantum polynomial-time algorithm for simon’s problem. *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*.
- [67] Lov K. Grover. Quantum computers can search rapidly by using almost any transformation. *Physical Review Letters*, 80(19):4329–4332, May 1998.
- [68] Lov K. Grover. Quantum computers can search arbitrarily large databases by a single query. *Physical Review Letters*, 79(23):4709–4712, Dec 1997.

-
- [69] Varun Garg Anupama P, Charles H. Bennett, and Ibm Thomas. Optimization of grover's search algorithm.
- [70] T. Loke and J.B. Wang. Efficient quantum circuits for szegedy quantum walks. *Annals of Physics*, 382:64 – 84, 2017.