

Resultados

Contents

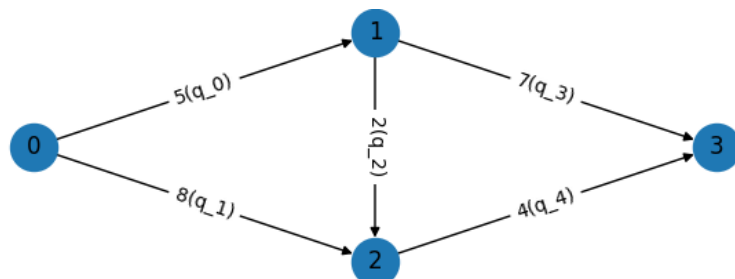
1	Notación	1
2	Primer grafo	3
2.1	Aer - Versión del paper (primer_grafo/aer-qaoa.ipynb)	4
2.1.1	Caso correcto	5
2.1.2	Caso erróneo	6
2.1.3	Caso subóptimo	7
2.1.4	Utilizando el parámetro theta obtenido en el artículo . .	8
2.1.5	Rz(*2), Rzz(*2), Rx(*2)	9
2.1.6	coef *= 2	9
2.1.7	Coef /= 2	9
2.1.8	$\beta \neq 2$	10
2.1.9	$\gamma \neq 2$	11
2.1.10	$\beta \neq 2, \gamma \neq 2$	12
2.1.11	$\beta_0 = 0.5, \gamma_0 = 0.5$	13
2.1.12	Original pero variar num layers	14
2.2	Aer simulator con restricción extra (primer_grafo/con_restricc/aer-qaoa.ipynb)	14
2.2.1	Caso correcto	15
2.2.2	Caso "correcto" con ruido	16
2.3	Provider	17

1 Notación

fun = Mínimo local hallado de la función *execute_circuit* con el optimizador
p = Número de capas (a mayor número el circuito es más profundo)
theta = Lista de parámetros $[\beta_1, \dots, \beta_p, \gamma_1, \dots, \gamma_p]$ del circuito cuántico
num iterations = Número de iteraciones del compilador necesarias para hallar el mínimo
seed_simulator = Semilla utilizada en la ejecución del circuito para fijar la aleatoriedad en *backend.run()*
X_{ij} = Se refiere a la arista **i** -> **j**. 1 Si dicha arista es parte del camino resultante, 0 en otro caso

$\mathbf{q_n}$ = Qubit enésimo
 $q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$

2 Primer grafo



2.1 Aer - Versión del paper (primer_grafo/aer-qaoa.ipynb)

Pruebas realizadas sobre la versión del código sin la restricción

$$\mathbf{X}_{13} + \mathbf{X}_{23} = \mathbf{1}$$

Versión equivalente a la de [Multi-Objective Routing Optimization for 6G Communication Networks Using a Quantum Approximate Optimization Algorithm-sensors-22-07570-v2]

- **Estadísticas:**

Realizando la ejecución 1000 veces se han obtenido como caminos resultantes los siguientes:

Qubits	Camino	Frecuencia (1000)
10101	$X_{01}X_{12}X_{23}$	917
10110	$X_{02}X_{12}X_{23}$	82
01001	$X_{01}X_{13}$	1

2.1.1 Caso correcto

fun	theta	num iterations	seed_simulator
29.63	[0.7739, 0.9302]	29	10

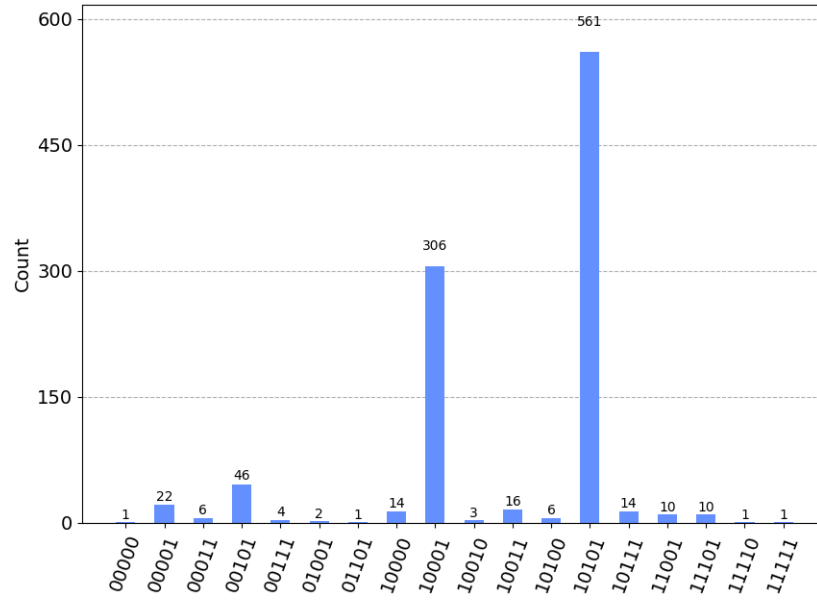


Figure 1: seed_simulator=10

Mejor resultado: 10101 ($q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$)
Camino: $X_{01}X_{12}X_{23}$ (Camino óptimo)

2.1.2 Caso err3neo

fun	theta	num iterations	seed_simulator
52.79	[0.6320 0.7177]	35	21

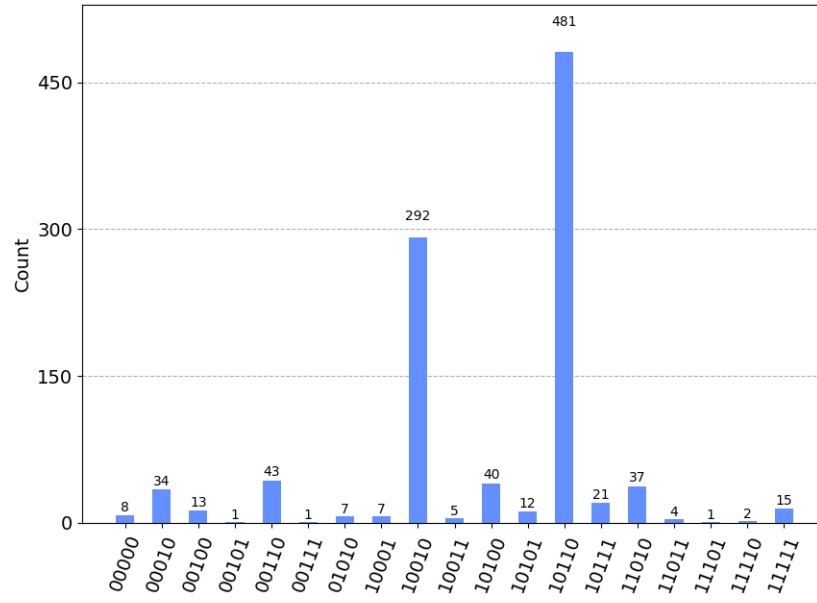


Figure 2: seed_simulator=21

Mejor resultado: 10110 ($q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$)

Camino: $X_{02}X_{12}X_{23}$ (Camino incorrecto. Rompe 2 restricciones)

Restricciones rotas:

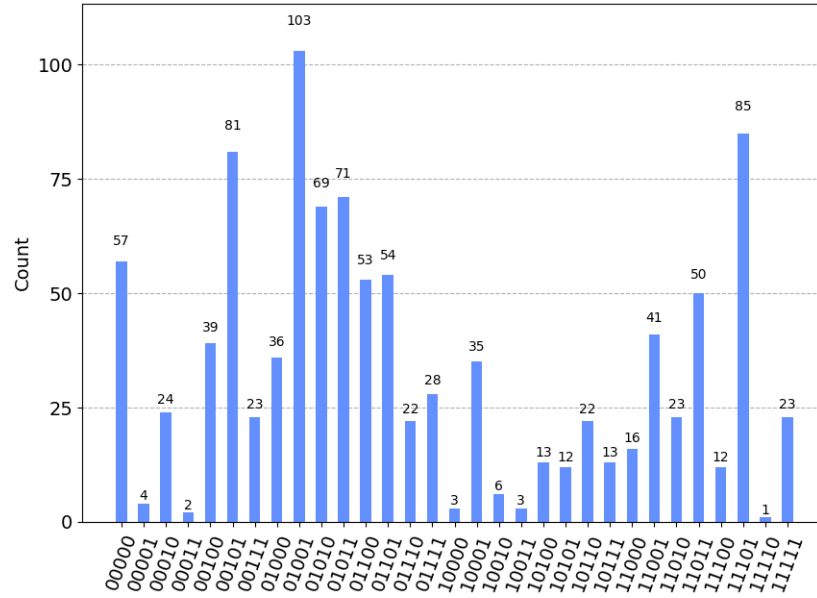
$$X_{02} + X_{12} = X_{23}$$

$$X_{01} = X_{12} + X_{13}$$

2.1.3 Caso subóptimo

Obtenido a mano (no se ha encontrado ninguna semilla que diese este resultado)

fun	theta
67.33	[-0.4811, 1.566]

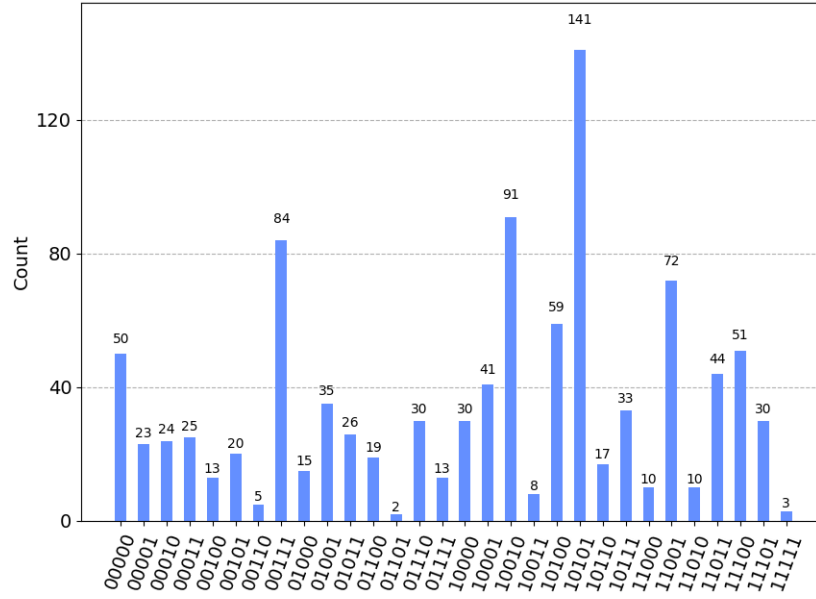


Mejor resultado: 01001 ($q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$)

Camino: $X_{01}X_{13}$ (Camino subóptimo, pero no se rompe ninguna restricción)

2.1.4 Utilizando el parámetro theta obtenido en el artículo

fun	theta
65.40	[0.28517317, -5.05969577]



Mejor resultado: 10101 ($q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$)

Camino: $X_{01}X_{12}X_{23}$ (Camino óptimo)

La gráfica resultante es muy similar a la versión que se intenta replicar. **fun** tiene resultados muy altos, entre 65 y 70 (en comparación con la versión del código con la restricción extra).

2.1.5 $R_z(*2)$, $R_{zz}(*2)$, $R_x(*2)$

```
circuit.rz(coef * 2, q_idx)
circuit.rzz(coef * gamma[p] * 2, q_idx[0], q_idx[1])
circuit.rx(beta[p] * 4, q_idx)
```

Qubits	Camino	Frecuencia (1000)
11010	$X_{02}X_{13}X_{23}$	845
11001		88
01010		5
11011		14
00101		21
00010		1
10110		14
10101		3
01001		5
10010		2
00110		2

2.1.6 $\text{coef} = 2$

```
circuit.rz(2 * coef, q_idx)
circuit.rzz(2 * coef * gamma[p], q_idx[0], q_idx[1])
circuit.rx(beta[p] * 2, q_idx)
```

Qubits	Camino	Frecuencia (1000)
11010	$X_{02}X_{13}X_{23}$	966
11101		2
00101		18
01010		8
11001		3
00110		1
10110		2

Da un mismo error un porcentaje de veces muy alto. Error muy fiable.

2.1.7 $\text{Coef} \neq 2$

```
circuit.rz(1/2 * coef, q_idx)
circuit.rzz(1/2 * coef * gamma[p], q_idx[0], q_idx[1])
circuit.rx(beta[p] * 2, q_idx)
```

Qubits	Camino	Frecuencia (1000)
00000		1000

2.1.8 $\beta \neq 2$

```
circuit.rz(coef, q_idx)
circuit.rzz(coef * gamma[p], q_idx[0], q_idx[1])
circuit.rx(beta[p], q_idx)
```

Qubits	Camino	Frecuencia (1000)
10101	<u>Óptimo</u>	986
10110		14

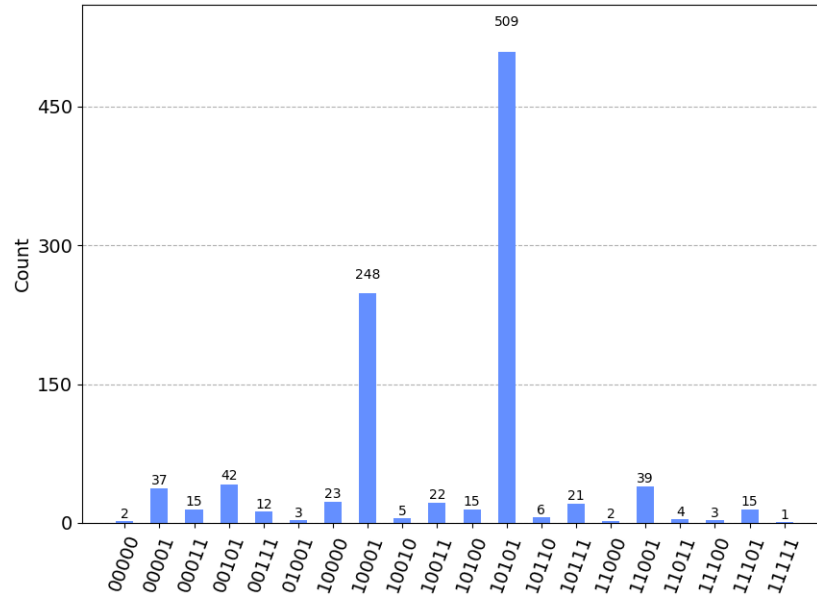


Figure 3: Mejor resultado

2.1.9 $\gamma \neq 2$

```
circuit.rz(coef, q_idx)
circuit.rzz(coef * gamma[p] / 2, q_idxs[0], q_idxs[1])
circuit.rx(beta[p] * 2, q_idx)
```

Qubits	Camino	Frecuencia (1000)
10101	<u>Óptimo</u>	1000

2.1.10 $\beta \neq 2, \gamma \neq 2$

```
circuit.rz(coef, q_idx)
circuit.rzz(coef * gamma[p] / 2, q_idx[0], q_idx[1])
circuit.rx(beta[p], q_idx)
```

- num layers = 1:

Qubits	Camino	Frecuencia (1000)
10101	<u>Óptimo</u>	1000

- num layers = 2:

Qubits	Camino	Frecuencia (1000)
10101		960
10001		28
11001		12

- num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101		565
10001		111
11101		87
...		...

2.1.11 $\beta_0 = 0.5, \gamma_0 = 0.5$

- num layers = 1:

Qubits	Camino	Frecuencia (1000)
10101		1000

- num layers = 2:

Qubits	Camino	Frecuencia (1000)
10101		992
11001		8

- num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101		469
11101		198
11011		88
...		...

2.1.12 Original pero variar num layers

Al aumentar el número de capas se obtienen resultados mucho peores (tal vez esté mal implementado)

- num layers = 1: (Igual que tabla de estadísticas normal)

Qubits	Camino	Frecuencia (1000)
10101		913
10110		86
01001		1

- num layers = 2:

Qubits	Camino	Frecuencia (1000)
10101		646
10010		75
10110		70
10001		49
...		...

- num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101		634
10010		92
10001		84
01001		66
00000		36

2.2 Aer simulator con restricción extra (primer_grafo/con_restricc/aer-qaoa.ipynb)

Con respecto a la función de coste del paper se añade la restricción

$$\mathbf{X}_{13} + \mathbf{X}_{23} = \mathbf{1}$$

Esto sería, que el camino solo llegue al nodo final **3** por una de las aristas X_{i3} existentes.

- **Estadísticas:**

Realizando la ejecución 1000 veces se han obtenido como caminos resultantes los siguientes:

Qubits	Camino	Frecuencia (1000)
10101	$X_{01}X_{12}X_{23}$	938
11000	$X_{13}X_{23}$	37
10001	$X_{01}X_{23}$	9
00011	$X_{01}X_{02}$	11
00100	X_{12}	3
00010	X_{02}	1
11111	$X_{01}X_{02}X_{12}X_{13}X_{23}$	1

2.2.1 Caso correcto

fun	theta	num iterations	seed_simulator
42.29	[0.5081, 0.9401]	33	3

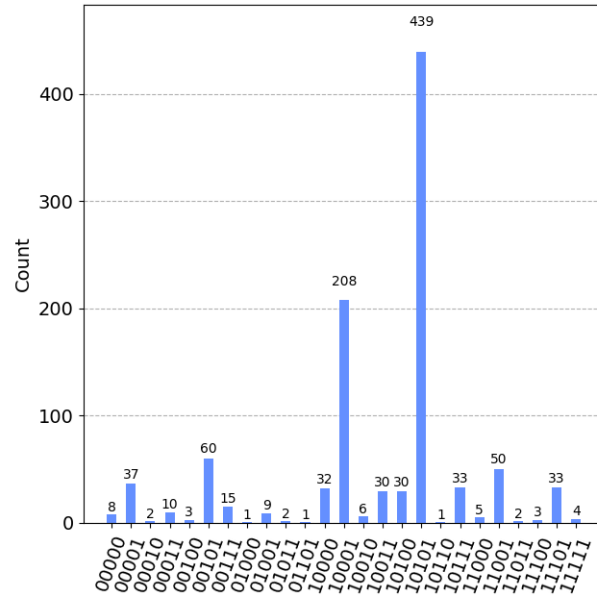


Figure 4: seed_simulator=3

Mejor resultado: 10101 ($q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$)
Camino: $X_{01}X_{12}X_{23}$ (Camino óptimo)

2.2.2 Caso "correcto" con ruido

fun	theta	num iterations	seed_simulator
90.75	[0.9962, 1.995]	27	2

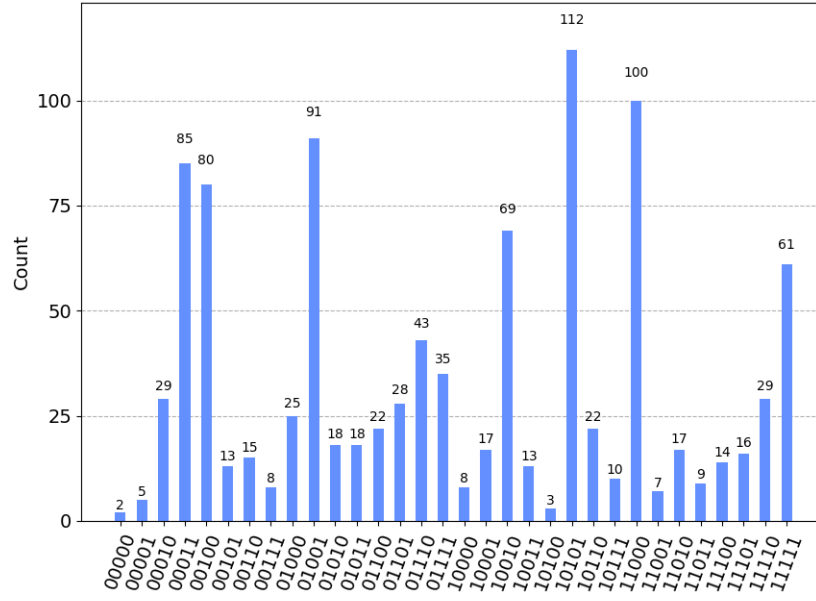


Figure 5: seed_simulator=2

Mejor resultado: 10101 ($q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$)

Camino: $X_{01}X_{12}X_{23}$ (Camino óptimo)

Aunque se obtenga el resultado óptimo (10101) existen otros resultados demasiado altos, e incluso ejecutando el circuito con el mismo **theta** se dan valores distintos. Podría afectar a los resultados de las estadísticas.

Además se ve que encuentra un valor **fun** demasiado alto (90.75)

2.3 Provider

- ibmq_lima

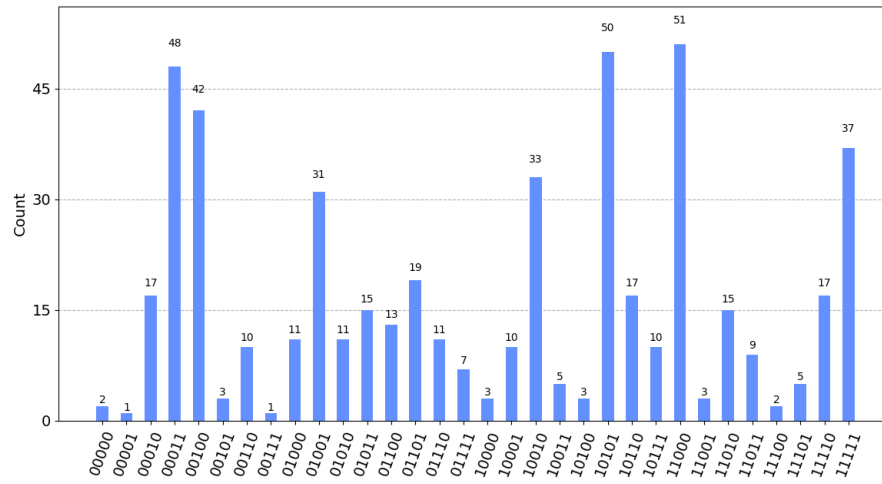


Figure 6: num iterations=2

Solo para comprobar que funciona la ejecución.