

# Resultados

## Contents

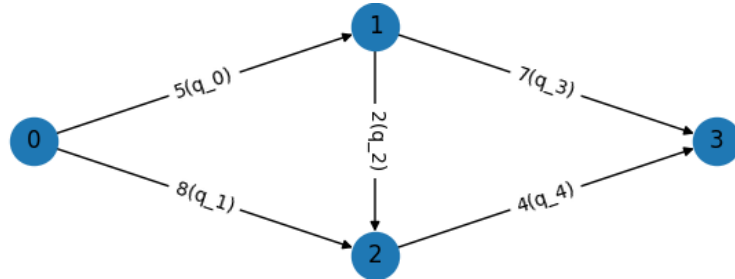
<b>1</b>	<b>Notación</b>	<b>2</b>
<b>2</b>	<b>Primer grafo sin restricción</b>	<b>3</b>
2.1	Aer - Versión del paper (primer_grafo/aer-qaoa.ipynb)	4
2.1.1	Caso correcto	5
2.1.2	Caso erróneo	6
2.1.3	Caso subóptimo	7
2.1.4	Utilizando el parámetro <b>theta</b> obtenido en el artículo	8
2.1.5	Rz(*2), Rzz(*2), 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 sin restricción extra y $P = 40$	15
2.2.1	Modificadores paper	15
2.2.2	Modificadores originales	17
2.3	Provider	19
2.3.1	ibmq_lima	19
2.3.2	ibmq_manila	20
2.4	Runtime	21
2.4.1	ibmq_lima	21
<b>3</b>	<b>Primer grafo con restricción</b>	<b>22</b>
3.1	Aer simulator con restricción extra (primer_grafo/con_restriccc/aer-qaoa.ipynb)	22
3.1.1	Caso correcto num layers = 1	23
3.1.2	Caso "correcto" con ruido num layers = 1	24
3.1.3	Gamma function	25
3.2	Aer simulator con restricción extra y $P = 40$	25
3.2.1	Modificadores del paper	25
3.2.2	Modificadores originales	27
3.2.3	Conclusiones y Gamma function	27

3.3	Aer simulator con restricción extra y $P = 54$	29
3.3.1	Gamma function	30
<b>4</b>	<b>Grafo Zhiqiang (Con aer)</b>	<b>31</b>
4.1	Modificadores originales	31
4.1.1	$P=20$	31
4.1.2	$P=40$	33
4.1.3	Conclusiones	35
4.2	Modificadores del paper	36
4.2.1	$P=20$	36
4.2.2	$P=40$	38
4.2.3	Conclusiones	40
<b>5</b>	<b>DWave</b>	<b>41</b>
5.1	Primer grafo	41
5.2	Segundo grafo	42
5.3	Zhiqiang grafo	43

## 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  $i \rightarrow j$ . 1 Si dicha arista es parte del camino resultante, 0 en otro caso  
 **$q_n$**  = Qubit enésimo  
 $q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$

## 2 Primer grafo sin restricción



## 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

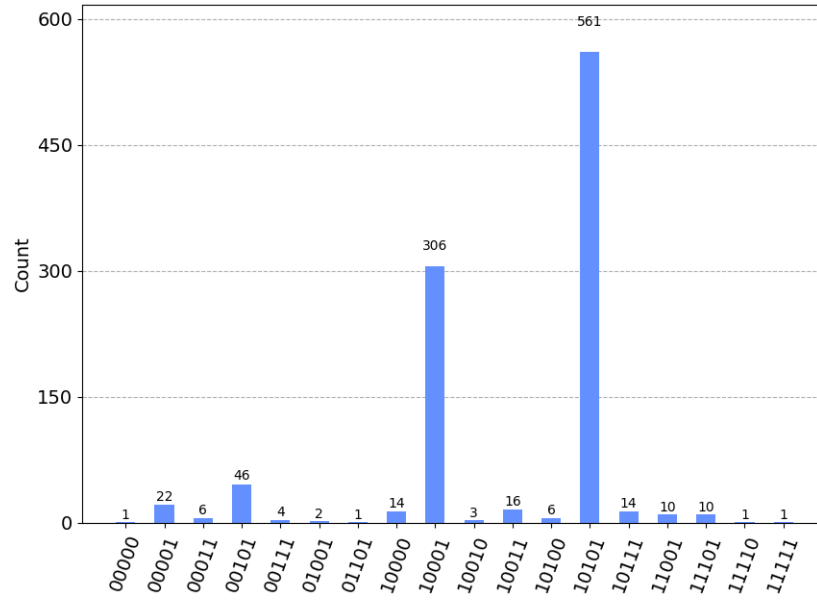


Figure 1: seed\_simulator=10

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

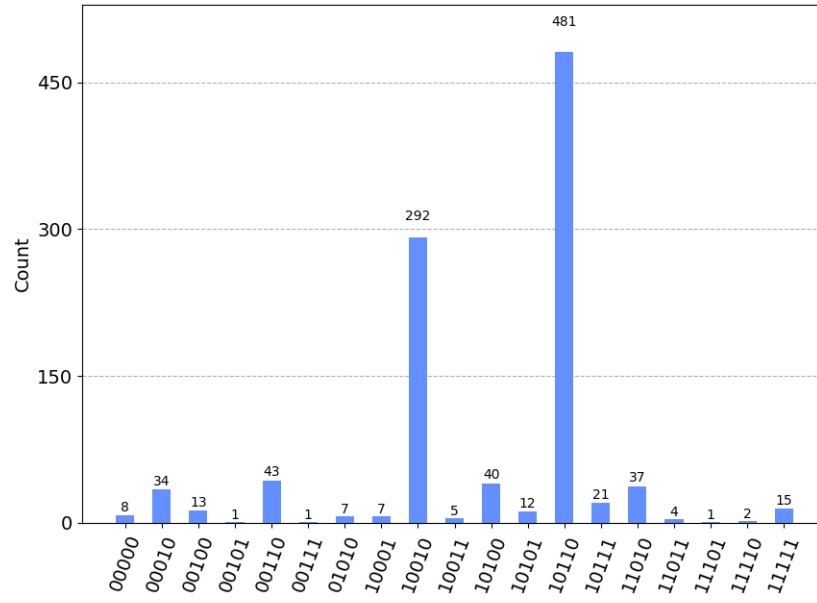


Figure 2: seed\_simulator=21

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:

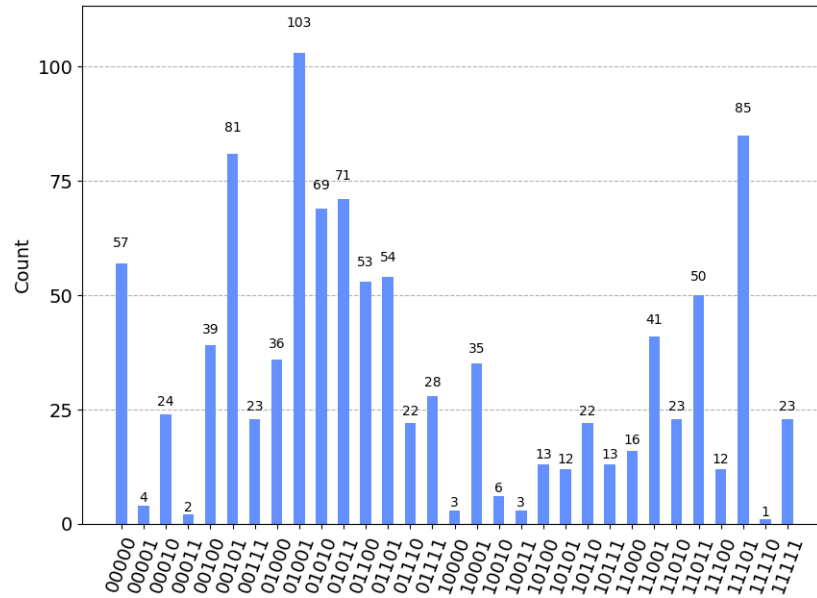
$$X_{02} + X_{12} = X_{23}$$

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

### 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]

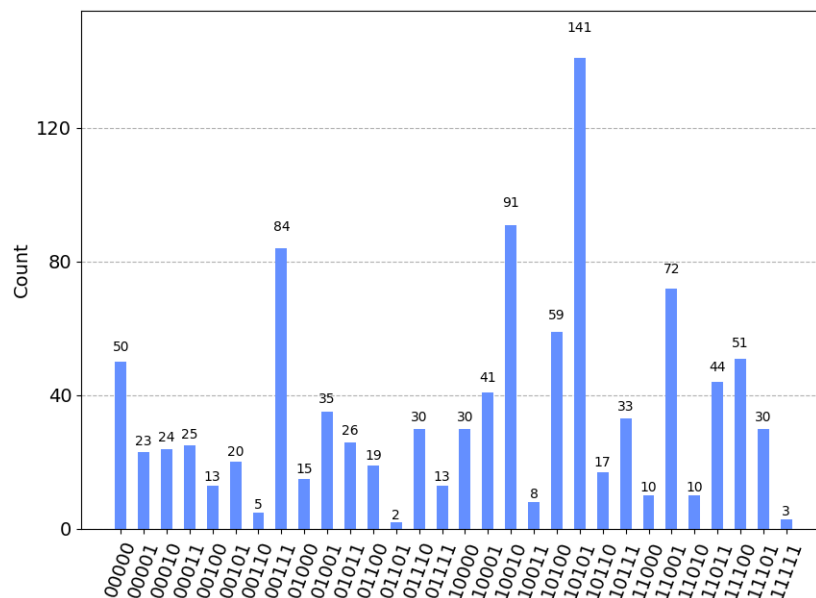


Mejor resultado: 01001 ( $q_4q_3q_2q_1q_0 = X_{23}X_{13}X_{12}X_{02}X_{01}$ )

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

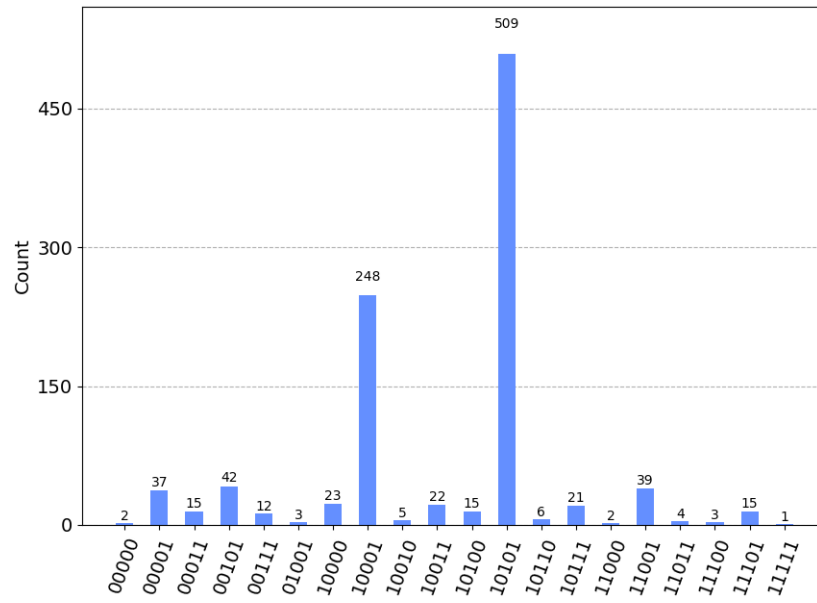


Figure 3: Mejor resultado

### 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:

<b>Qubits</b>	<b>Camino</b>	<b>Frecuencia (1000)</b>
10101		1000

- num layers = 2:

<b>Qubits</b>	<b>Camino</b>	<b>Frecuencia (1000)</b>
10101		992
11001		8

- num layers = 3:

<b>Qubits</b>	<b>Camino</b>	<b>Frecuencia (1000)</b>
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 sin restricción extra y $P = 40$

### 2.2.1 Modificadores paper

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

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
10010	<b>Óptimo</b>	780
10001		200
10101		20

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
10010	<b>Óptimo</b>	766
10101		166
10111		23
10001		10
10110		9
01001		9
11010		4
00001		4
11011		4
00000		3
10011		1
01101		1

- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	396
01001		196
01101		154
10010		105
00101		38
10001		32
10110		31
00001		21
11001		5
...		...

- Estadísticas num layers = 4:

<b>Qubits</b>	<b>Camino</b>	<b>Frecuencia (1000)</b>
01001	<b>Óptimo</b>	484
01101		198
10101		96
10010		65
00001		47
11001		40
01010		13
11101		12
10110		9
00101		8
00010		5
10001		5
00000		4
11011		4
11010		3
10011		2
10100		2
01000		1
11000		1
01011		1



### 2.2.2 Modificadores originales

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

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	403
10010		146
11001		103
00000		97
01001		95
10000		74
00001		39
10001		21
00101		10
10011		5
11011		4
11101		2
11010		1

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	194
00000		157
10010		120
01001		72
10110		59
00001		48
10000		23
11010		35
10001		21
00010		43
01101		10
11011		66
11101		21
01000		16
10100		40
10111		10
10011		8
11000		4
00101		18
01011		5
00011		3
11001		20
11111		1
00100		4
01010		2

- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
10010	<b>Óptimo</b>	444
01001		206
10101		137
00000		51
11101		29
...		...

## 2.3 Provider

### 2.3.1 ibmq\_lima

Solo para comprobar que funciona la ejecución.

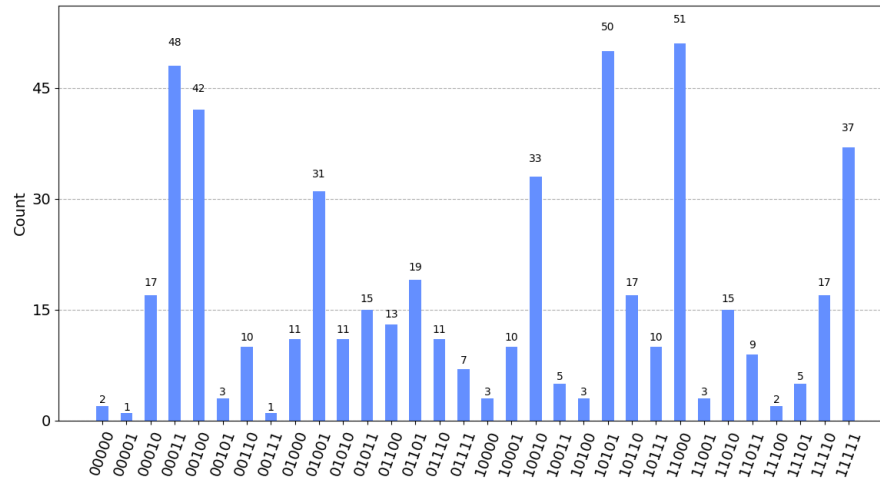
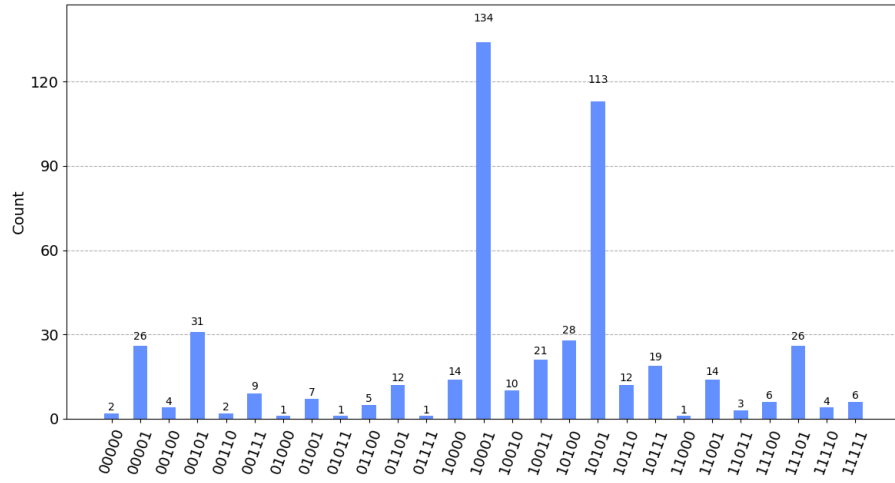


Figure 4: num iterations=2

### 2.3.2 ibmq\_manila

$\beta_0 = 0.5, \gamma_0 = 0.5$



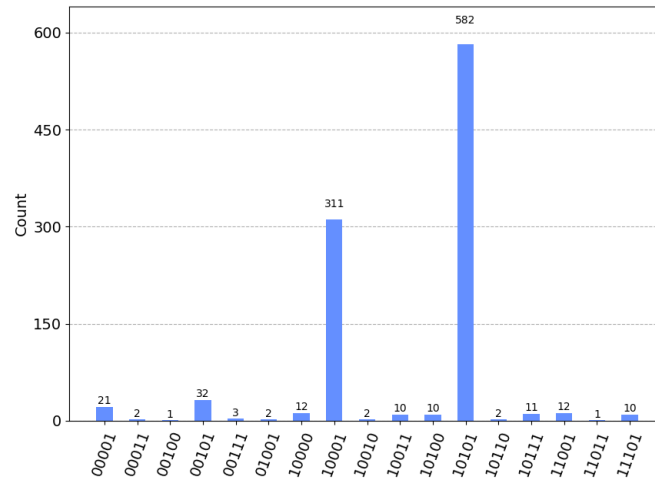
## 2.4 Runtime

### 2.4.1 ibmq\_lima

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

fun	theta	num iterations
37.16	[0.6869, 0.4728]	26

Resultado de ejecutar ese **theta** obtenido con Aer:



### 3 Primer grafo con restricción

#### 3.1 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

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

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

Modificadores del paper

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	938
11000		37
10001		9
00011		11
00100		3
00010		1
11111		1

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	646
01010		159
10001		58
01001		42
10010		28
00100		19
11011		18
...		...

- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	848
10001		67
10010		25
01001		16
01010		10
10110		6
11101		4
...		...

### 3.1.1 Caso correcto num layers = 1

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

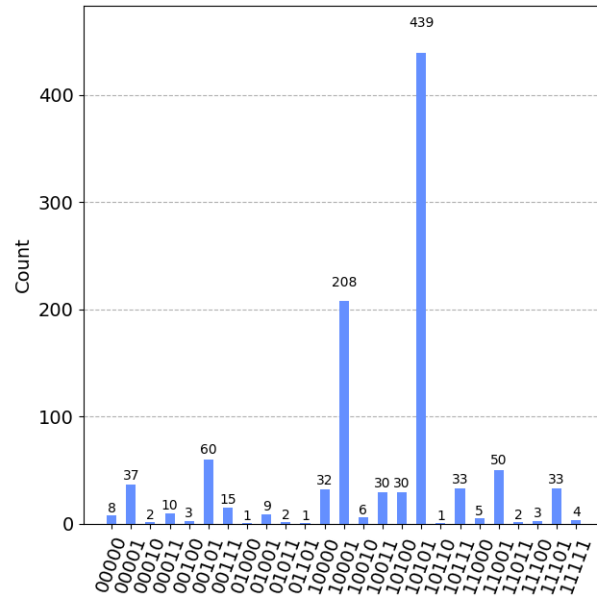


Figure 5: seed\_simulator=3

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)

### 3.1.2 Caso "correcto" con ruido num layers = 1

Table 1: captionada

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

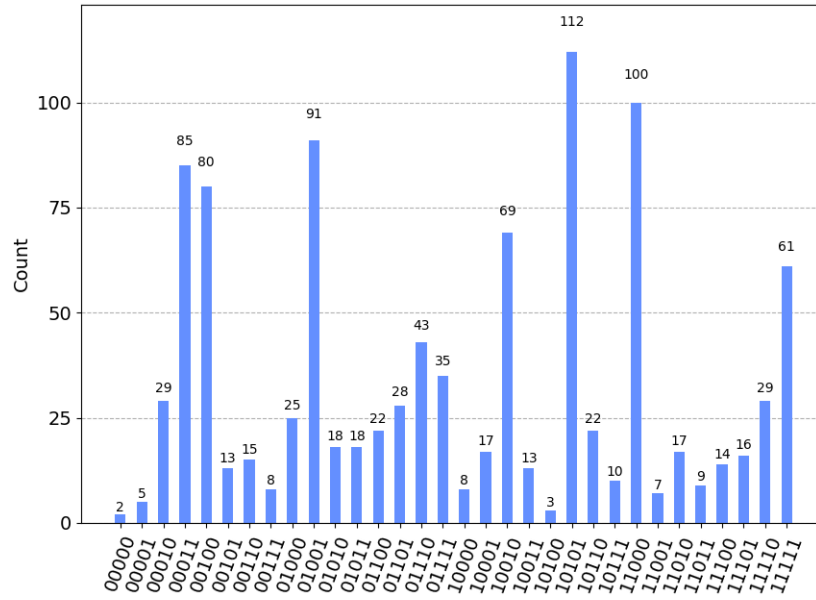


Figure 6: seed\_simulator=2

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)

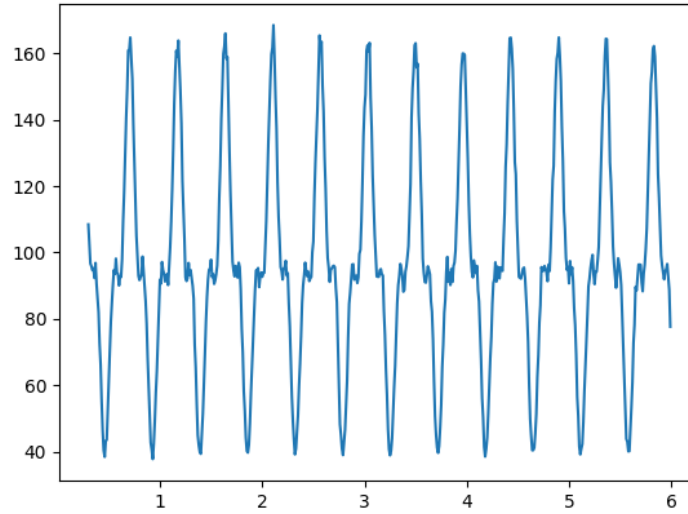
Aunque se obtenga el resultado óptimo (10101) existen otros resultados demasiado altos, e incluso ejecutando el circuito con el mismo **theta** se dan valores distintos. Podría afectar a los resultados de las estadísticas.

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



### 3.1.3 Gamma function

Variación de `execute_circuit` con respecto a  $\gamma$  con `num layers = 1` y  $\beta = 1.0$



## 3.2 Aer simulator con restricción extra y $P = 40$

### 3.2.1 Modificadores del paper

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

- Estadísticas `num layers = 1`:

Qubits	Camino	Frecuencia (1000)
10001	<b>Óptimo</b>	825
10101		174
01010		1

- Estadísticas `num layers = 2`:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	544
10001		152
01010		94
01001		84
10010		62
11001		27
10110		15
...		...

- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	693
01001		69
10001		65
10010		42
01010		31
00011		26
01011		20
11001		14
00010		10
...		...

### 3.2.2 Modificadores originales

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

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	630
00001		139
11001		99
10001		39
10011		36
01001		28
10010		14
...		...

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	382
10001		167
01001		105
10010		73
00000		43
...		...

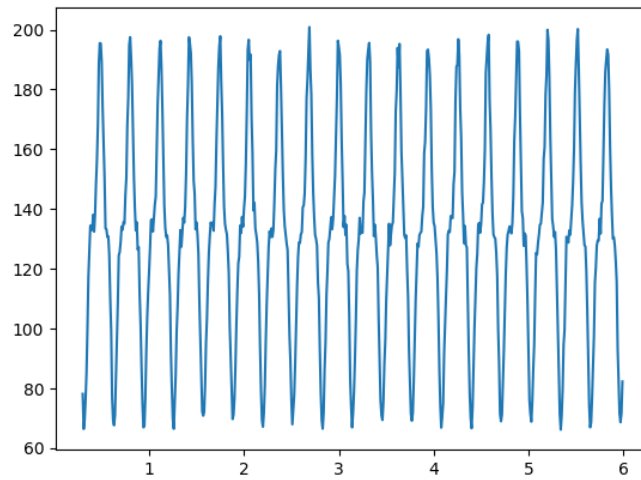
- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
10101	<b>Óptimo</b>	412
01001		134
10010		93
10001		83
00001		81
...		...

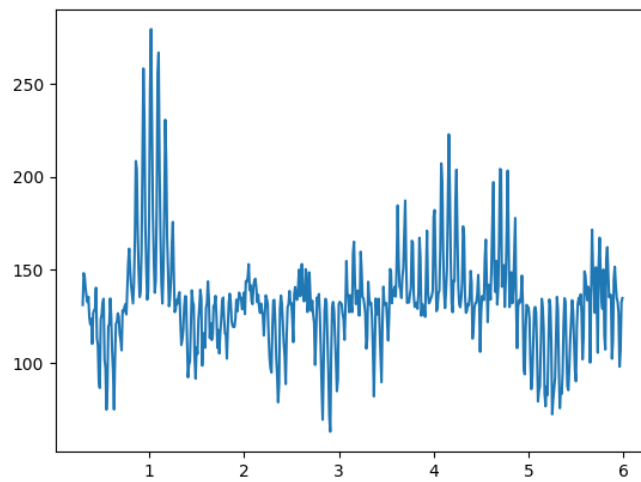
### 3.2.3 Conclusiones y Gamma function

- Para ambos modificadores los resultados sí parecen mejorar con el número de capas (aunque den resultados muy bajos)

1. Modificadores del paper



## 2. Modificadores originales



### 3.3 Aer simulator con restricción extra y $P = 54$

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

Probar a aumentar el factor de penalización para las restricciones ( $P = 54$ ).

- Estadísticas num layers = 1:

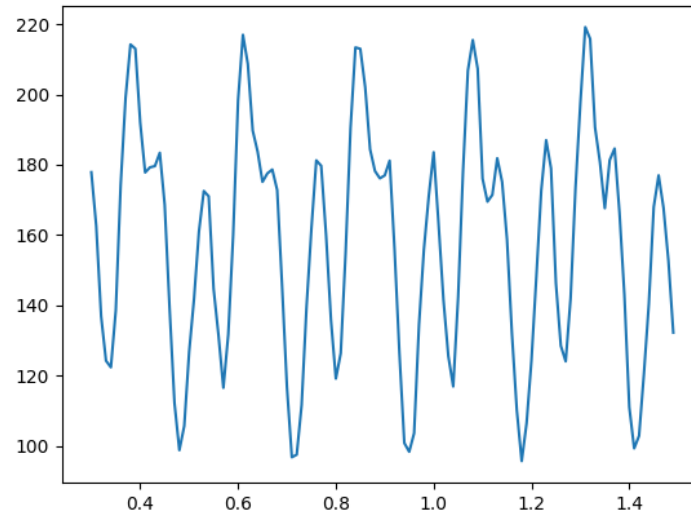
Qubits	Camino	Frecuencia (1000)
10010	<b>Óptimo</b>	251
01010		236
01001		222
10101		135
10001		100
00001		46
01101		4
10110		3
00101		1
00000		1
10111		1

- Estadísticas num layers = 2:

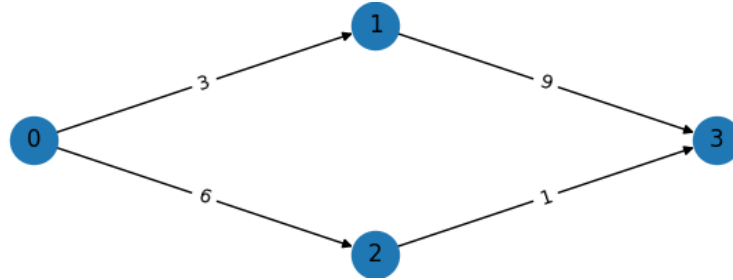
Qubits	Camino	Frecuencia (1000)
01010	<b>Óptimo</b>	280
10010		199
01001		190
10101		183
10001		85
00001		37
01101		13
10110		7
11010		2
00000		2
11011		1
00010		1

### 3.3.1 Gamma function

Variación de `execute_circuit` con respecto a  $\gamma$  con `num layers` = 1 y  $\beta = 1.0$



## 4 Grafo Zhiqiang (Con aer)



### 4.1 Modificadores originales

```

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

```

#### 4.1.1 P=20

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
0101	$X_{01}X_{13}$	912
0110		86
1010	$X_{02}X_{23} \rightarrow \text{Óptimo}$	1
0111		1

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
0101		965
0000		22
1010	<b>Óptimo</b>	8
0001		2
1001		1
0110		1
1000		1

- Estadísticas num layers = 4:

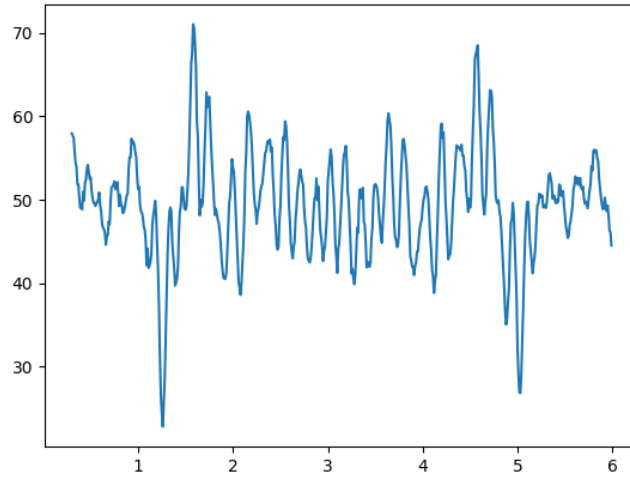
Qubits	Camino	Frecuencia (1000)
0101		985
1010	<b>Óptimo</b>	15

- Estadísticas num layers = 10:

Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	687
0101		308
0000		2
0001		1
0100		1
1110		1
1110		1

1. Gamma function

Variación de execute\_circuit con respecto a  $\gamma$  con num layers = 1 y  $\beta = 1.0$





#### 4.1.2 P=40

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
1000	<b>Óptimo</b>	260
1010		202
1001		147
0101		141
1101		66
1011		59
1111		29
0001		23
0100		20
0010		17
0000		13
0110		12
0111		7
1110		4

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	481
0101		215
1001		62
0000		55
0010		50
1000		31
0100		26
1101		22
1011		17
1110		16
0001		8
0110		6
0111		6
1111		5

- Estadísticas num layers = 3:

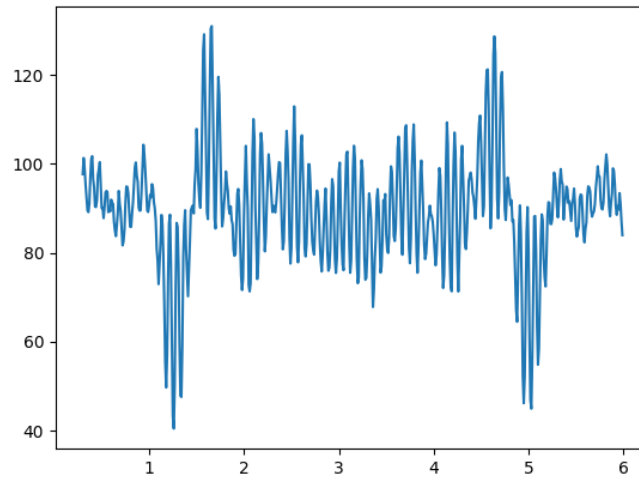
Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	513
0101		387
1000		33
1101		21
1001		21
1110		6
0000		5
1111		4
1011		4
0111		3
0010		2
0100		1

- Estadísticas num layers = 10:

Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	564
0101		431
0100		2
0111		1
1110		1
0000		1

1. Gamma function

Variación de execute\_circuit con respecto a  $\gamma$  con num layers = 1 y  $\beta = 1.0$



### 4.1.3 Conclusiones

- A diferencia de "primer grafo" con  $P=20$  los resultados mejoran al aumentar el número de capas, aunque siguen siendo muy malos resultados.
- Los resultados de la función con respecto a  $\gamma$  son muy ruidosos, puede ser el motivo de tan malos resultados.
- Al utilizar  $P=40$  se obtiene una gráfica más ruidosa y con valores mucho más altos al romper restricciones. Esto es de esperar, ya que  $P$  se utiliza como factor de penalización en la función de coste  $(P^*(\dots))^2$ .
- El aumento de  $P$  puede estar provocando que resultados de función cercanos al óptimo sean menos probables (en las gráficas con  $P=20$  hay un mínimo  $\sim 20$  y con  $P=40$  es  $\sim 40$ )

## 4.2 Modificadores del paper

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

### 4.2.1 P=20

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
1101	<b>Óptimo</b>	381
1001		182
0010		178
1011		97
0110		44
0100		44
0001		26
...		...
1010		4
...		...

- Estadísticas num layers = 2:

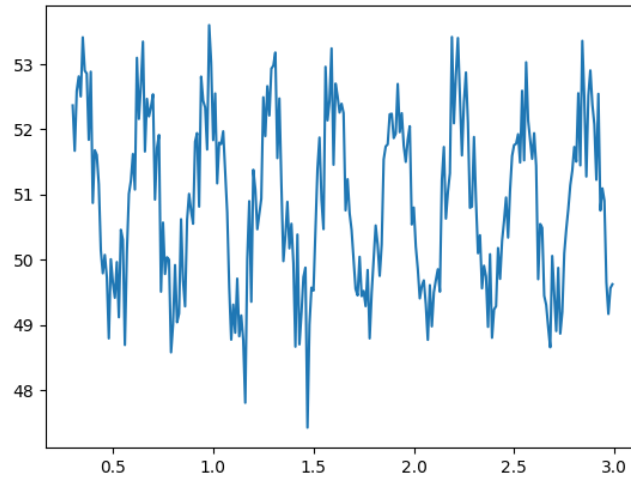
Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	324
0101		285
1011		268
1001		102
1110		20
0100		1

- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	928
0101		68
1011		4

1. Gamma function

Variación de `execute_circuit` con respecto a  $\gamma$  con `num layers` = 1 y  $\beta = 1.0$



#### 4.2.2 P=40

- Estadísticas num layers = 1:

Qubits	Camino	Frecuencia (1000)
1101		714
0100		124
0010		47
0110		85
1000		2
1001		11
1011		10
1110		2
0111		1
1111		1
0001		2
0000		1

El óptimo ni siquiera aparece (**0%**)

- Estadísticas num layers = 2:

Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	392
0100		182
0110		148
0010		139
...		...

- Estadísticas num layers = 3:

Qubits	Camino	Frecuencia (1000)
1010	<b>Óptimo</b>	637
0101		183
1101		71
1011		42
0100		18
0110		13
1001		11
1110		9
0010		7
0000		4
0001		3
1000		2

- Estadísticas num layers = 4:

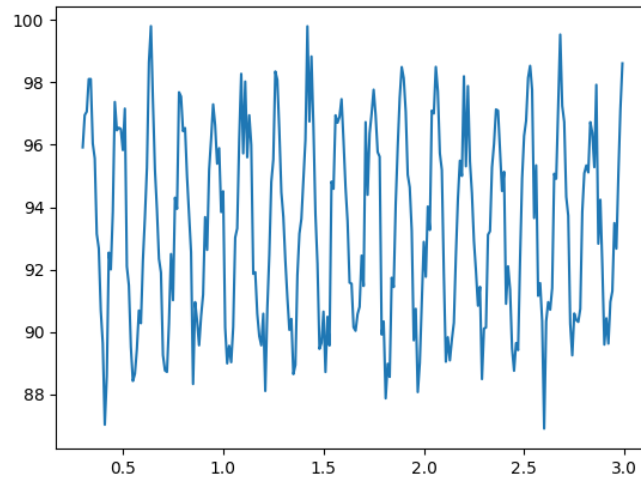
Qubits	Camino	Frecuencia (1000)
0101	<b>Óptimo</b>	661
1010		337
0100		1
1101		1

- Estadísticas num layers = 10:

Qubits	Camino	Frecuencia (1000)
0101	<b>Óptimo</b>	568
1010		426
1000		1
0111		1
0010		1
0000		2
1101		1
1101		1

# 1. Gamma function

Variación de execute\_circuit con respecto a  $\gamma$  con num layers = 1 y  $\beta = 1.0$



### 4.2.3 Conclusiones

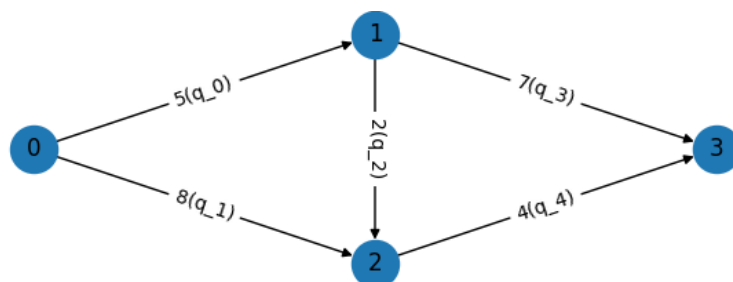
- Mejora muy clara con el aumento de capas (con  $P=20$  y 3 capas se acierta un **92.8%**).



## 5 DWave

Ejecuciones con 1024 medidas

### 5.1 Primer grafo

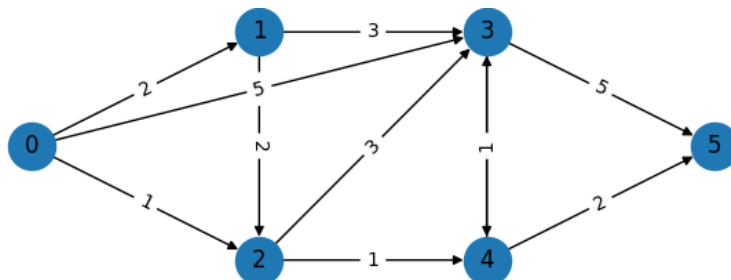


Sampleset:

	(0, 1)	(0, 2)	(1, 2)	(1, 3)	(2, 3)	energy	num oc.	chain .
0	1	0	1	0	1	11.0	348	0.0
1	0	1	0	0	1	12.0	373	0.0
2	1	0	0	1	0	12.0	294	0.0
3	0	0	0	0	0	54.0	4	0.0
4	0	0	0	0	1	58.0	1	0.0
5	1	0	0	0	0	59.0	1	0.0
6	0	0	1	0	1	60.0	1	0.0
7	1	0	1	0	0	61.0	1	0.0
8	0	1	0	1	0	69.0	1	0.0

['BINARY', 9 rows, 1024 samples, 5 variables]

## 5.2 Segundo grafo

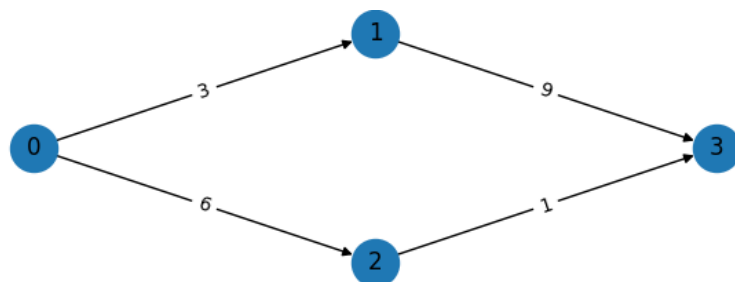


Sampleset:

	(0, 1)	(0, 2)	(0, 3)	(1, 2)	(1, 3)	(2, 3)	...	(4, 5)	energy	num oc.	...
0	0	1	0	0	0	0	...	1	4.0	72	...
1	0	1	0	0	0	0	...	1	6.0	69	...
2	0	1	0	0	0	1	...	1	7.0	65	...
3	1	0	0	1	0	0	...	1	7.0	37	...
134	0	1	0	0	0	1	...	1	7.0	1	...
4	0	0	1	0	0	0	...	1	8.0	51	...
5	0	1	0	0	0	0	...	0	8.0	92	...
6	1	0	0	0	1	0	...	1	8.0	42	...
7	0	1	0	0	0	1	...	0	9.0	47	...
8	1	0	0	1	0	0	...	1	9.0	34	...
9	1	0	0	0	1	0	...	0	10.0	29	...
10	1	0	0	1	0	1	...	1	10.0	33	...
11	0	0	1	0	0	0	...	0	10.0	39	...
12	0	1	0	0	0	1	...	0	11.0	53	...
13	1	0	0	1	0	0	...	0	11.0	29	...
14	1	0	0	0	1	0	...	0	12.0	35	...
15	1	0	0	1	0	1	...	0	12.0	32	...
16	0	0	1	0	0	0	...	0	12.0	54	...
17	1	0	0	1	0	1	...	0	14.0	42	...
20	0	1	0	0	0	0	...	0	55.0	3	...
...	...	...	...	...	...	...	...	...	...	...	...

['BINARY', 137 rows, 1024 samples, 11 variables]

### 5.3 Zhiqiang grafo



Sampleset:

	(0, 1)	(0, 2)	(1, 3)	(2, 3)	energy	num oc.	chain .
0	0	1	0	1	7.0	776	0.0
1	1	0	1	0	12.0	247	0.0
2	0	1	0	0	46.0	1	0.0