

Project3 Solving SAT Problem by Quantum Circuit

組別: Group23

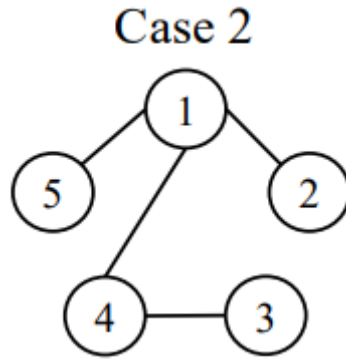
組員: N26134992 賴郁明、M16131111 童品綸

content

1.	Quantum circuit for solving graph coloring problem	2
1.1	Encode problem.....	2
1.2	apply the phase oracle.....	3
1.3	apply the diffuser.....	3
1.4	simulation result.....	4

1. Quantum circuit for solving graph coloring problem

1.1 Encode problem



我們選取題目中的 Case 2 來使用 Quantum circuit 解決著色問題，該 case 題目如上圖，我們以 x_i 變數來假設每個頂點的顏色，其中 i 代表頂點名稱，我們先以兩個顏色進行著色，並判斷是否符合 SAT。其中：

$i \in \{1,2,3,4,5\}$: 表示頂點集合 V

$$x_i = \begin{cases} 1, & \text{如果頂點 } i \text{ 被著色為顏色 1} \\ 0, & \text{如果頂點 } i \text{ 被著色為顏色 0} \end{cases}$$

因此，需要符合的規則有：

- 相鄰頂點的顏色不同

相鄰的情況有以下節點，若相鄰則兩個節點不能是相同的顏色，當 x_i, x_j 兩者相鄰時，不可相等， $x_i \neq x_j$ ，故可得出以下幾組邏輯關係。

(1,2) : $(\neg x_1 \vee \neg x_2)$ 和 $(x_1 \vee x_2)$

(3,4) : $(\neg x_3 \vee \neg x_4)$ 和 $(x_3 \vee x_4)$

(1,5) : $(\neg x_1 \vee \neg x_5)$ 和 $(x_1 \vee x_5)$

(1,4) : $(\neg x_1 \vee \neg x_4)$ 和 $(x_1 \vee x_4)$

將這些式子 encode 成該著色問題的 CNF，如下式：

$$(\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_5) \wedge (x_1 \vee x_5) \wedge (\neg x_1 \vee \neg x_4) \wedge (x_1 \vee x_4)$$

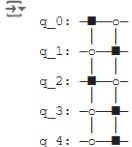
代入下解可以使上式為 true，故該式滿足 SAT 條件，因此僅需兩種顏色進

行著色。

$$x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0$$

1.2 apply the phase oracle

```
1 cnf_formula = "(~x1 | ~x2) & (x1 | x2) & (~x3 | ~x4) & (x3 | x4) & (~x1 | ~x5) & (x1 | x5) & (~x1 | ~x4) & (x1 | x4) "
```



我們是利用 qiskit 的 PhaseOracle 函式將我們輸入的 CNF 轉成一個量子電路，當輸入量子位對應到「滿足公式」的解時，就對那個狀態翻轉相位 (phase)；不滿足時則不動。

1.3 apply the diffuser

1.配置量子模擬器：

```
1 # Configure backend
2 backend = Aer.get_backend('aer_simulator')
3 quantum_instance = QuantumInstance(backend, shots=1024)
```

使用 Qiskit 的 Aer 模擬器作為後端來執行量子電路。shots=1024 表示執行 1024 次量子測量以獲取結果的分佈。

2. 定義問題：

```
5 # Create a new problem from the phase oracle and the
6 # verification function
7 problem = AmplificationProblem(oracle=oracle, is_good_state=v.is_correct)
```

- 將前面構建的 oracle 和驗證函數 is_correct 包裝為一個 AmplificationProblem。
- oracle 用於標記滿足 CNF 公式的量子態。
- is_good_state 驗證某量子態是否為解，其中的 v.is_correct 是使用我們定義的 Verifier class 來判斷該 assignment 是否滿足給定的 CNF，下圖為其使用範例。

```
1 v = Verifier(cnf_formula)
2 v.is_correct('00001')

False
```

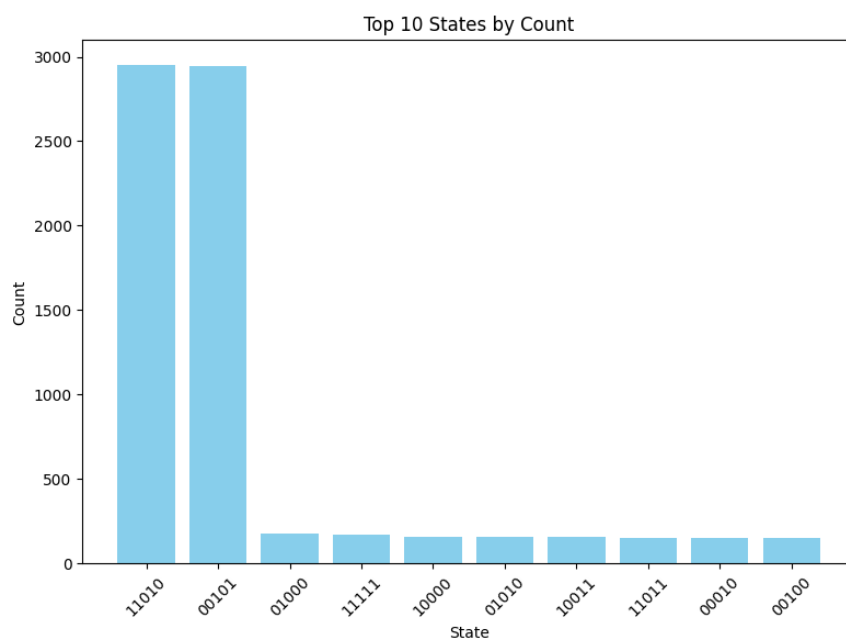
3. 應用 Grover 演算法：

```
9 # Use Grover's algorithm to solve the problem
10 grover = Grover(quantum_instance=quantum_instance)
11 result = grover.amplify(problem)
12 result.top_measurement
```

- 使用 Grover 演算法的量子電路來放大「解態」的振幅。
- 透過多次疊加 Oracle 和擴散閘 (Diffuser) 的應用來達成效果。
- 最後從量子測量結果中提取出測量次數最多的量子態，這個態即為 CNF 公式的解。

1.4 simulation result

最後模擬結果將最常出現的 10 個狀態印成長條圖，可以看到明顯 11010，00101 的出現頻率遠大於其他，須注意圖表上方解的順序為(x5,x4,x3,x2,x1)，將這兩組解輸入先前的 Verifier class 確認結果滿足該 CNF 之 SAT，故滿足此著色問題。



```
1 v.is_correct('10100')
⇒ True

[25] 1 v.is_correct('01011')
⇒ True
```

根據公式計算出最大的 Grover iterations 為 3

$$N_{\max} = \frac{\pi}{4 \sin^{-1} \left(\sqrt{\frac{2}{32}} \right)}$$

$$N_{\max} = 3.10826$$

下圖為解出的兩種著色方式:

