

初期状態が不完全なグローバーのアルゴリズムの振る舞いについて

9BSP1118 村岡海人

2023 年 1 月 10 日

目次

1	はじめに	2
1.1	研究の背景	2
1.2	研究の目的	2
2	基本的内容	2
2.1	量子ビット	2
2.1.1	単一量子ビット	2
2.1.2	多量子ビット	3
2.2	量子計算	4
2.3	量子アルゴリズム	4
2.4	グローバーのアルゴリズム	4
2.4.1	概要	4
2.4.2	アルゴリズムの流れ	5
2.4.3	図を使用しての説明	6
2.4.4	最適な k の見積もり	8
3	本論	9

1 はじめに

1.1 研究の背景

量子コンピュータとは、量子力学を利用して計算を行うコンピュータである。この量子コンピュータで行う計算を量子計算と呼び、量子計算におけるアルゴリズムのことを量子アルゴリズムと呼ぶ。例えば、多項式時間で整数を因数分解するショアのアルゴリズムや、整列化されていないデータベースからデータベースから特定のデータを探索するグローバーのアルゴリズムがある。

1.2 研究の目的

従来の計算機が論理演算から構成されているのと同様、量子計算も量子演算から構成されており、この量子演算は、時間に依存するシュレディンガー方程式から記述することができる。量子計算を行う際に、ハミルトニアンや時間がズレてしまうと実現したい操作からズレた操作を行うことになり、アルゴリズム自体の出力に対するエラーになってしまう。本研究では、初期状態を準備する操作が不完全な場合に、グローバーのアルゴリズムがどれだけ機能するか調べることを目的とした。

2 基本的内容

2.1 量子ビット

ビットは古典計算と古典情報の基本概念である。量子計算と量子情報は類似の概念である量子ビットの上に構築される。

2.1.1 単一量子ビット

まず、量子ビットの説明をする。古典ビットに1あるいは0の状態に対応した、状態 $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ と $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ がある。ここで、量子状態を表すために、ケット記号 ($| \rangle$) を使ったディラックの記法を用意した。量子ビットと古典ビットの違いは、量子ビットは $|0\rangle$ と $|1\rangle$ の重ね合わせ状態を取り得ることである。これは次のように $|0\rangle$ と $|1\rangle$ の線型結合として、

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (2.1)$$

と表される。ここで、 α, β は複素数であり、複素確率振幅と呼ぶ。量子ビットの状態は2次元複素ベクトル空間のベクトルで表される。特に $|0\rangle$ と $|1\rangle$ 計算基底状態と呼び、この2次元複素ベク

トル空間の正規直交基底を構成する。

古典計算では、古典ビットを調べてそれが 0, 1 のいずれの状態にあるかを定めることができる。例えば、コンピュータがメモリの内容を取り出す時にいつもこれを行なっている。量子ビットは量子ビットを調べてその量子状態、つまり、 α と β の値を決めることはできない。量子ビットに対して $|0\rangle$ と $|1\rangle$ のいずれの状態にあるかを調べる測定を行うと、確率 $|\alpha|^2$ で $|0\rangle$ 、確率 $|\beta|^2$ で $|1\rangle$ が得られる。全確率の和は 1 なので、 $|\alpha|^2 + |\beta|^2 = 1$ である。幾何学的解釈ではこれは量子ビットの状態が長さ 1 に正規化される条件である。したがって、一般的に量子ビットの状態は 2 次元複素ベクトル空間の単位ベクトルを表す。

量子ビットは自由度が 2 の多くの系で実現されている。例えば、核スピン、単一光子の 2 つの異なる偏光、単一原子における電子軌道の 2 つの状態などがある。原子モデルで電子は基底状態、または励起状態に存在しそれぞれを $|0\rangle, |1\rangle$ と呼ぶ。

次のような幾何学的表現が量子ビットを考える上で有用な描像である。 $|\alpha|^2 + |\beta|^2 = 1$ であるので、式 (2.1) を次のように書き換える。

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \quad (2.2)$$

ここで、 θ, φ は実数である。図に示すように、 θ, φ は 3 次元単位球面上の点を定義する。この球面をブロッホ球と呼ぶ。

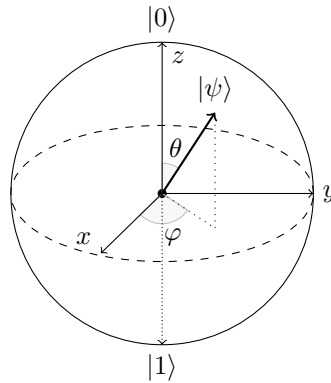


図 1 量子ビットのブロッホ球表示

これは単一量子ビット状態を視覚化する便利な方法である。単一量子ビットの操作はブロッホ球上の描像で記述できる。しかし、ブロッホ球は多量子ビットに対して一般化できないことに注意する。

2.1.2 多量子ビット

多量子ビットの状態について考えてみる。簡単のため、2 個の量子ビットがあるとする。これが古典ビットならば 4 つの取り得る状態 00, 01, 10, 11 がある。これに対して 2 個の量子ビットの系

には $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ で表される計算基底状態がある。2 個の量子ビットを記述する状態ベクトルは、

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \quad (2.3)$$

で与えられる。ここで、 $\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}$ はそれぞれの基底の複素確率振幅である。単一量子ビットの場合と同様に、測定結果 $x(= 00, 01, 10, 11)$ は確率 $|\alpha_x|^2$ で生じ、測定後の量子ビットの状態は $|x\rangle$ となる。確率の合計が 1 になる条件は正規化状態 $\sum_{x \in \{0,1\}^2} |\alpha_x|^2 = 1$ で表される。ここで、記号「 $\{0,1\}^2$ 」は「各文字が 0 または 1 であり、長さ 2 の記号列の集合」を意味する。

一般に n 個の量子ビットを考えると、この系の計算基底は $|x_1 x_2 \cdots x_n\rangle$ の形をしており、この系の量子状態は 2^n 個の振幅で規定される。ここで、 $x = x_1 x_2 \cdots x_n$ は、 $x \in \{0,1\}^n$ であり、 $x \in \{0,1\}^n$ は各文字が 0 または 1 であり、長さ n の記号列の集合を表す。

2.2 量子計算

多量子ビットの状態について考えてみる。簡単のため、2 個の量子ビットがあるとする。これが古典ビットならば 4 つの取り得る状態 $00, 01, 10, 11$ がある。これに対して 2 個の量子ビットの系には $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ で表される計算基底状態がある。2 個の量子ビットを記述する状態ベクトルは、

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \quad (2.4)$$

で与えられる。ここで、 $\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}$ はそれぞれの基底の複素確率振幅である。単一量子ビットの場合と同様に、測定結果 $x(= 00, 01, 10, 11)$ は確率 $|\alpha_x|^2$ で生じ、測定後の量子ビットの状態は $|x\rangle$ となる。確率の合計が 1 になる条件は正規化状態 $\sum_{x \in \{0,1\}^2} |\alpha_x|^2 = 1$ で表される。ここで、記号「 $\{0,1\}^2$ 」は「各文字が 0 または 1 であり、長さ 2 の記号列の集合」を意味する。

一般に n 個の量子ビットを考えると、この系の計算基底は $|x_1 x_2 \cdots x_n\rangle$ の形をしており、この系の量子状態は 2^n 個の振幅で規定される。ここで、 $x = x_1 x_2 \cdots x_n$ は、 $x \in \{0,1\}^n$ であり、 $x \in \{0,1\}^n$ は各文字が 0 または 1 であり、長さ n の記号列の集合を表す。

2.3 量子アルゴリズム

2.4 グローバーのアルゴリズム

本節では、データベース探索など、いわゆる探索問題を解く量子アルゴリズムを説明する。量子探索アルゴリズムは古典の探索アルゴリズムより計算量が少なく、高速であると言われている。

2.4.1 概要

このグローバーのアルゴリズムは以下のような流れで行う。 N 個のデータに対して $O(\sqrt{N})$ 回の計算量で解を見出すことができる。古典的な探索アルゴリズムでも同じ計算量を持つ 2 分探索アルゴリズムがあるが、2 分探索アルゴリズムは事前にソートされているデータを扱うため、ソートされていないデータの探索アルゴリズムではグローバーのアルゴリズムの方が高速である。

このグローバーのアルゴリズムは以下のような流れで行う。 n を量子ビット数とすると、 $N = 2^n$ の要素からなるデータベースから M 個の解を探索する問題を考え、要素のラベルを n 桁のビット列 $x = x_1 \cdots x_n$ とする。

- 全ての状態の重ね合わせ状態 $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ を用意する
- 演算子 U_w (解に対する反転操作) を作用させる
- 演算子 $U_s(|s\rangle)$ を対象軸にした反転操作) を作用させる
- 2、3 を k 回繰り返す

2.4.2 アルゴリズムの流れ

まず初めに、全ての状態の重ね合わせ状態 $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$ を用意する。初期状態 $|0\rangle^{\otimes n} = |0 \cdots 0\rangle$ に対して全ての量子ビットにアダマール演算子を作用させると、

$$\begin{aligned}
 |s\rangle &= H^{\otimes n} |0\rangle^{\otimes n} \\
 &= (H \otimes \cdots \otimes H) |0 \cdots 0\rangle \\
 &= \frac{1}{\sqrt{2^n}} (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle) \\
 |s\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1}
 \end{aligned} \tag{2.5}$$

のように計算できる。

次に解に対する反転操作を作用させる。入力 $|x\rangle$ に対して x が解なら、 -1 をかけて位相を反転し、解でないならば 1 をかける。つまり、単一量子ゲートを以下のように定義する。

$$\begin{cases} U_w |x\rangle = |x\rangle & (x \neq w) \\ U_w |w\rangle = -|w\rangle \end{cases} \tag{2.6}$$

$$U_w = I - 2 \sum_{w \in \text{解}} |w\rangle \langle w| \tag{2.7}$$

w は検索したい値である。これを用いると、 $|s\rangle$ は、

$$\begin{aligned}
 U_w &= \frac{1}{\sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} U_w |x\rangle + \frac{1}{\sqrt{2^n}} U_w |w\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle - \frac{1}{\sqrt{2^n}} |w\rangle \\
 U_w |s\rangle &= |s\rangle - \frac{2}{\sqrt{2^n}} |w\rangle
 \end{aligned} \tag{2.8}$$

のように計算できる。

最後に、 $|s\rangle$ を対象軸にした反転操作 U_s を定義する。

$$\begin{cases} U_s |x\rangle = 2 \langle s | x | s \rangle - |x\rangle = \frac{2}{\sqrt{2^n}} |x\rangle \\ U_s |s\rangle = 2 \langle s | s | s \rangle - |s\rangle = |s\rangle \end{cases} \quad (2.9)$$

$$U_s = 2 |s\rangle \langle s| - I \quad (2.10)$$

式 (2.9) より、 U_s を作用させると、

$$\begin{aligned} U_s U_w |s\rangle &= |s\rangle - \frac{2}{\sqrt{2}} \left(\frac{2}{\sqrt{2^n}} |s\rangle - |w\rangle \right) \\ &= \frac{2^n - 4}{2^n} |s\rangle + \frac{2}{\sqrt{2^n}} |w\rangle \\ &= \frac{2^n - 4}{2^n \sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle + \left(\frac{2^n - 4}{2^n \sqrt{2^n}} + \frac{2}{\sqrt{2^n}} \right) |w\rangle \\ U_s U_w |s\rangle &= \frac{2^n - 4}{2^n \sqrt{2^n}} \sum_{x=0, x \neq w}^{2^n-1} |x\rangle + \frac{3 \cdot 2^n - 4}{2^n \sqrt{2^n}} |w\rangle \end{aligned} \quad (2.11)$$

のように計算できる。

$|s\rangle$ の時の状態では、 $|w\rangle$ を測定すると、確率は $\frac{1}{2^n}$ となる。式 (2.11) から確率が上昇していることがわかる。この確率を増幅させる操作のことを、反復増幅と呼ぶ。グローバーのアルゴリズムは、この反復増幅を複数かい行うことにより、 $|w\rangle$ の確率を 1 に近づける。

2.4.3 図を使用しての説明

$|w\rangle$ に直行するベクトル $|w^\perp\rangle$ を用いた平面を考えると、以下のような状態が得られる。

$$|w\rangle = \frac{1}{\sqrt{N-M}} \sum_{x=0, w \neq 0}^{2^n-1} |x\rangle \quad (2.12)$$

$$|w^\perp\rangle = \frac{1}{\sqrt{M}} |w\rangle \quad (2.13)$$

全ての状態の重ね合せ状態 $|s\rangle$ は次のように表すことができるので、2次元平面ベクトルであることがわかる。

$$|s\rangle = \sqrt{\frac{N-M}{N}} |w^\perp\rangle + \sqrt{\frac{M}{N}} |w\rangle \quad (2.14)$$

全ての状態の重ね合わせ状態 $|s\rangle$ は次のように表せるので、この2次元平面内ベクトルであることがわかる。式 (2.14) より、 $\cos \frac{\theta}{2} = \sqrt{\frac{N-M}{N}}$, $\sin \frac{\theta}{2} = \sqrt{\frac{M}{N}}$ を満たす角 θ を用いれば、と表すことができる。これを図示すると、図2.4.3のようになる。

$$|s\rangle = \cos \frac{\theta}{2} |w^\perp\rangle + \sin \frac{\theta}{2} |w\rangle \quad (2.15)$$

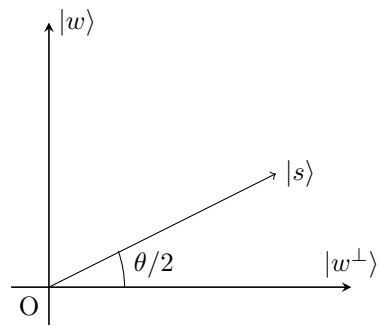


図 2 全ての状態の重ね合わせ状態 $|s\rangle$

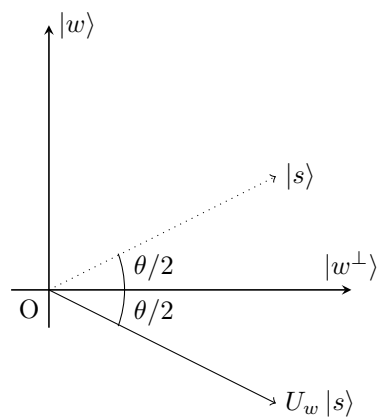


図 3 $|s\rangle$ に U_w を作用

次に、 $|s\rangle$ に U_w をかけることにより、 $|w^\perp\rangle$ を軸に反転すると、図??のようになる。

最後に、 U_s を作用させることにより、 $|s\rangle$ を軸に $U_w |s\rangle$ を反転させると、図3のようになる。

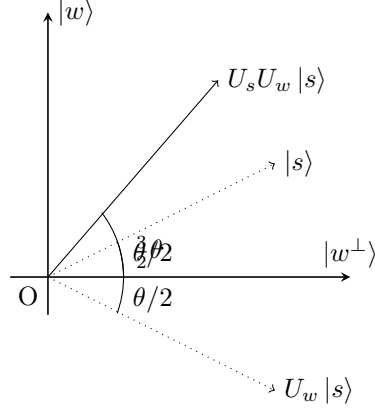


図4 $U_w |s\rangle$ に U_s を作用

以上より、平面ベクトル内で、角度 θ だけの回転が行われ、 $|w\rangle$ を測定する確率が上昇することがわかる。

2.4.4 最適な k の見積もり

最後に、 $U_s U_w$ を作用させる回数 k について、最適な回数が幾つなのか調べる。式 (3.2) より、グローバーのアルゴリズムを1回施すと、

$$U_s U_w |s\rangle = \cos \frac{3}{2} \theta |w^\perp\rangle + \sin \frac{3}{2} \theta |w\rangle \quad (2.16)$$

となる。これを k 回施すと、

$$(U_s U_w)^k |s\rangle = \cos \frac{2k+1}{2} \theta |w^\perp\rangle + \sin \frac{2k+1}{2} \theta |w\rangle \quad (2.17)$$

となる。これを用いて、最終的に $|w\rangle$ の確率振幅を1にしたいので、

$$\begin{aligned} \sin \frac{2k+1}{2} \theta &= 1 \\ \Leftrightarrow \frac{2k+1}{2} \theta &= \frac{\pi}{2} \\ k &= \frac{\pi}{2\theta} \end{aligned} \quad (2.18)$$

となる。よって、 $\frac{2k+1}{2} \theta$ が $\frac{\pi}{2}$ にもっとも近くなるときは、

$$R = \text{ClosestInteger}\left(\frac{\pi}{2\theta} - \frac{1}{2}\right) \quad (2.19)$$

の時である。ここで、 $\text{ClosestInteger}(\dots)$ は \dots に最も近い整数を表す。

最後に、 R の上限を評価する。 $\theta > 0$ について成り立つ式、

$$\frac{\theta}{2} \geq \sin \frac{\theta}{2} = \sqrt{\frac{M}{N}} \quad (2.20)$$

を使うと、以下のように表すことができる。

$$R \leq \left(\frac{\pi}{2\theta} - \frac{1}{2} \right) + 1 = \frac{\pi}{2\theta} + \frac{1}{2} \leq \frac{\pi}{4} \sqrt{\frac{N}{M}} + \frac{1}{2} \quad (2.21)$$

つまり、 R は $O(\sqrt{N/M})$ である。これにより、グローバーのアルゴリズムが $O(\sqrt{N})$ で動作することがわかる。

3 本論

これは本論の中身です。