量子コンピューティングを用いたグラフ彩色問題の解法比較

井出朋美　(指導教員：工藤和恵)

1　はじめに

近年多くの企業で用いられている量子コンピュータは一部の問題において従来のコンピュータよりも高速な計算が可能と言われている。また、複雑な問題やシミュレーションに対して新たな解法を提供する可能性がある。以上のことより、量子コンピュータを応用することで、科学や技術の進歩に寄与できると期待されている。

本研究では、イジング模型や QUBO 模型を使用するアニーリング型と量子回路で表現するゲート型、2 つの手法を用いて、グラフ彩色についてそれぞれの手法の違いを実際に確かめる。

2　問題設定

本研究ではグラフの頂点彩色問題を考える。頂点彩色問題とは隣接する頂点同士を別の色で塗り分ける問題である。

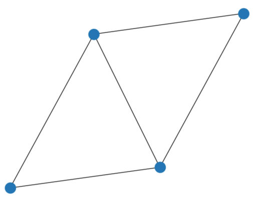

図 1: 本研究に用いるグラフ

図 1 のような 4 個のノードと 5 本の辺を持つグラフが与えられたとする。辺で繋がれたノード同士は異なる色となるように色分けをする。

3　アニーリング型

3.1　アニーリング型について

本研究では文献 [1] よりシミュレーテッドアニーリングを用いて実装する。本研究では二値変数が $\{\ 0, 1\ \}$ をとる QUBO 形式で定式化し、ハミルトニアンの最小値を求める。

3.2　定式化

定式化するとハミルトニアンは次式にようになる。

$$H = A\sum_{v=1}^{V}\left(1-\sum_{i=1}^{n}x_{v,i}\right)^2 + \sum_{(uv)\in E}\sum_{i=1}^{n}x_{u,i}x_{v,i} \quad (1)$$

$x_{v,i}$ はノード v が色 i で塗られた時に 1、その他では 0 とする。V が頂点の数、E が辺の集合、A が制約の強さ、n が色の数を表す。第一項が各ノードに 1 色だけ塗るという one-hot 制約を、第二項が辺で結ばれたノードの色が異なるという制約を表す。このハミルトニアンが 0 となる状態が見つかった時、グラフは n 色で塗ることが可能だとわかる。また、$x_{v,i}$ でどの i の時 1 となるかを調べることでノード v の色を調べることができる。

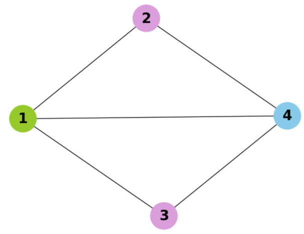

図 2: 彩色を施したグラフ

3.3　結果

$n = 3$ として実行したところ、図 2 のように隣り合うノードを違う色に塗ることができた。200 回実行し 197 回成功し、その平均時間は 0.42 秒となった。とても短い時間でほとんど成功させることができた。

4　ゲート型

ゲート型はアダマル (H) ゲート、回転ゲート、制御 NOT(CNOT) ゲートなどを用いて量子計算を行う。本研究では 2 種類のゲート型のアルゴリズムで頂点彩色を実行する。

4.1　QAOA について

QAOA(Quantum Approximate Optimazation Algorithm) は、量子近似最適化アルゴリズムを意味する。

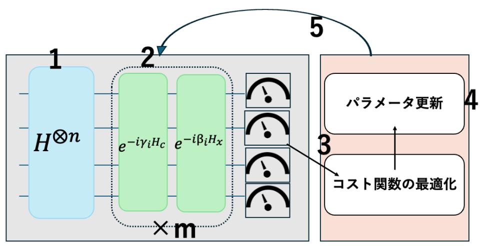

図 3: QAOA の手順 (文献 [2] の図を改変)

1. 初期状態 $|s\rangle$ として均等な重ね合わせ状態を作る (図 3 の 1 の操作)。
2. パラメータ β_i, γ_i $(i = 1, ..., m)$ を持つユニタリゲートを初期状態 $|s\rangle$ に m 回作用させる (図 3 の 2 の操作)。
3. 全量子ビットを測定してハミルトニアンの期待値を求める (図 3 の 3 の操作)。
4. 期待値が小さくなるように古典コンピュータでパラメータを更新する (図 3 の 4 の操作)。
5. 収束するまで手順 1 ～ 4 を繰り返し、最適解を求める (図 3 の 5 の操作)。

ここで図 3 の H_c はコスト関数を表したハミルトニアン、H_x は量子ゆらぎに対応して状態変化を起こすハミルトニアンである。

このアルゴリズムでは、パラメータを最適化するのに古典のオプティマイザを使用している。様々な種類を試したが、今回は制約つき最適化を行う COBYLA というオプティマイザの結果が最適だったためそれを使用する。実行の際に量子回路の実行回数 (shot 数) と図 3 の 2 の繰り返し回数 (m) が結果を大きく左右する。また今回は文献 [3] より初期状態として均一な重ね合わせ状態ではなく、one-hot 制約を満たす状態を使用した。

4.2 QAOA での結果

(a) (b)

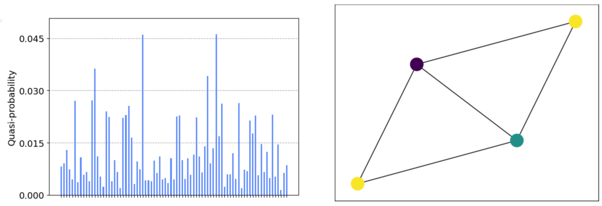

図 4: (a) 確率分布．(b) 成功した際のグラフ

図 4 の結果は shot 数 10000 回、$m = 13$ の時の結果である。それぞれの数が多いほど正確になるがその分時間もかかってしまう。この数は様々な数で試してみてその中でも成功する確率が高く、時間のかからない回数に設定している。200 回の実行中 129 回成功し、平均時間は 71.04 秒となった。

図 4(a) のグラフは確率分布で、横軸が彩色のパターンを示し、縦軸が各パターンの実現確率となっており、一番確率の高い解を図 4(b) のように結果として表示した。

4.3 Grover のアルゴリズムについて

ゲート型のもう一つの方法として文献 [4] より Grover のアルゴリズムを用いる。

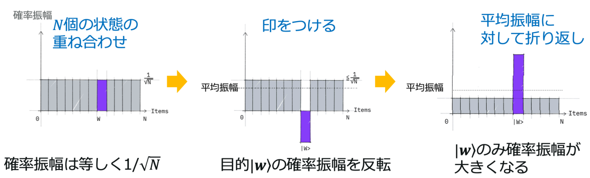

図 5: Grover のアルゴリズムの手順 (文献 [5] 引用)

1. QAOA と同様に重ね合わせ状態を作る (図 5 左)。
2. 目的の状態に印をつけ、この確率振幅を反転させる (図 5 中央)。
3. 平均振幅に対して折り返す量子回路を作成する。すると印のついた状態のみ確率振幅が大きくなる (図 5 右)。
4. 手順 2 と 3 を繰り返し、最適解を見つける。

この繰り返し回数は理論的に求めることができ、その回数は N 個のデータに関して M 個の正解がある場合 $\frac{\pi}{4}\sqrt{\frac{N}{M}} - \frac{1}{2}$ と見積もることができる。

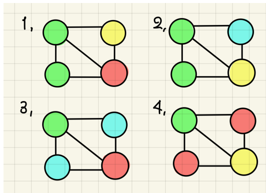

図 6: 上位 4 つのグラフ

4.4 Grover のアルゴリズムでの結果

図 6 は実現確率の上位 4 つの結果を表示している。今回は 5 回の繰り返しを行った。しかし図 6 の 3, 4 番目のみが正解だが、1, 2 番目のように不正解の結果も上位に出てきてしまった。200 回の実行中 117 回成功し、平均時間は 0.60 秒となった。今回扱ったアルゴリズムは 2 色塗り分けにしか対応していないため、成功率が下がってしまった。3 色対応にすると成功率は上がるが、量子ビットの数が増え、計算時間は数分かかってしまうと考えられる。

5 まとめ

結果はアニーリング型の方が計算時間が圧倒的に短く、結果も正しかった。グラフ彩色の様な組み合わせ最適化問題にはアニーリング型の方が効率がいいということがわかった。

QAOA は使用するオプティマイザや、初期状態の作り方、レイヤーの繰り返し数によってかかる時間や成功率も大きく異なるので安定しない。今後の課題として、プログラムを修正して精度を高める必要がある。Grover のアルゴリズムは時間はかかってしまうが成功率が上がるオラクルに修正する必要がある。すなわちアニーリング型と比べるとゲート型には問題を解ける計算量の限界が小さいと思われる。

参考文献

[1] OpenJij, Solving Graph ColoringProblem with PyQUBO `https://tutorial.openjij.org/build/html/ja/009-GraphColorPyqubo.html` (2024 年 1 月 30 日アクセス)

[2] Investor-daiki, QAOA アルゴリズムの量子回路の仕組みを理解する `https://www.investor-daiki.com/qaoa-circuit` (2024 年 1 月 30 日アクセス)

[3] Jij Tech Blog, JijModeling-Transpiler-Quantum で Quantum Alternating Operator Ansatz `https://zenn.dev/jij_inc/articles/jtq_graph_coloring#fn-74cf-3` (2024 年 1 月 30 日アクセス)

[4] Nathan Kjer, Quantum Computing: Map Coloring via Grover's Algorithm `https://nathankjer.com/grovers-algorithm/` (2024 年 1 月 30 日アクセス)

[5] IBM Research, IBM Research Tokyo `https://research.ibm.com/jp-ja/labs/tokyo/` (2024 年 1 月 30 日アクセス)