P-10

# パリティ関数の疎グラフへの埋め込みに関する実験的評価

**Experimental Evaluation of Mapping Parity Function Oracle onto Sparse Graphs**

○北村竜嗣 [1], 平石秀史 [2]
*Ryuji Kitamura[1], Hidefumi Hiraishi[2]

Abstract: In this research, we investigate the embedding overhead of implementing a parity function oracle into several sparse hardware graphs: grid, tree, and torus. We utilize the Deutsch-Jozsa algorithm as a benchmark problem and simulate its execution using Qiskit.

## 1. はじめに

Shor のアルゴリズムや Grover 探索に代表される量子アルゴリズムは，標準的な計算量理論的仮定の下で古典アルゴリズムの限界を超えた計算速度を達成することが理論的に示されており，新世代の計算機として量子計算機は注目を集めてきた．近年では量子計算機の開発競争が世界的に活発に行われており，実際に量子計算機を利用できるクラウドサービスを IBM や Amazon などが展開し始めている．

一方で，現状の量子デバイスの主流は NISQ（Noisy Intermediate-Scale Quantum）デバイスと呼ばれ計算過程でノイズの影響を強く受け，浅層回路しか事実上実行できないものとなっている．また，量子ビットも疎結合である場合が多く，全結合を前提とした量子アルゴリズムを実装する場合，回路の深さが増大してしまい効果的な計算が困難である．[1]

本稿では，Deutsch-Jozsa のアルゴリズムを例に，オラクルと呼ばれるブラックボックス関数の実装方法を具体化し，特定の量子ビットのトポロジーに埋め込んだ際のゲート数や深さについて評価を行う．Deutsch-Jozsa のアルゴリズムは，古典計算に対する量子計算の指数的加速の例が初めて確立されたアルゴリズムであり，量子鍵配送や暗号における応用可能性も指摘されている [2, 3].

## 2. 量子回路とその評価指標

先述の通り，NISQ においてはゲート数や回路の深さの増大が計算精度に大きな影響をもたらす．よって，現状の量子デバイスで効果的な計算を行うにはゲート数の小規模化や，回路の浅層化が重要である．特に回路の深さに関しては，誤り耐性の観点から重要視されており，古典的な情報処理を後から加えてノイズ効果を抑制する量子誤り抑制法では，量子回路の深さについて指数的な時間コストが必ず必要となることが示されている．[4]

## 3. 実験手法

量子ビットトポロジーは技術的な制約から平面グラフをはじめとした疎グラフである場合が多い．実際，IBM が公開している量子計算機では「heavy-hex lattice」という構造を採用しており，接続性を犠牲にエラーを抑制している．[5] 本研究では 4 種類のグラフ構造に対し，量子回路を埋め込んだときのゲート数及び，深さの変化を調べ，ノイズの影響評価を行う．Deutsch-Jozsa アルゴリズムにおけるオラクルは，1 であるビットの数の偶奇を表すパリティ関数とする．量子ビット数は 9 と 25，量子ビットのトポロジーは完全グラフ，トーラス，格子，二分木，使用できる量子ゲートは I，H，X，CNOT，SWAP に制限し，量子回路の埋め込み手法は qiskit に備わっているヒューリスティックなアルゴリズムを用いた．1 つの量子回路につき 1024 回処理を実行し，ビットパターンの分布を取得した．実行結果の評価指標は，観測値と理論値の間の $L^2$ ノルムを用いる．つまり，観測結果が理論値と近い程，評価指標の値は小さい値となる．

### 3.1 Deutsch-Jozsa アルゴリズム

Deutsch-Jozsa のアルゴリズムでは，入力として定数関数かバランス関数のいずれかが与えられる．

**定義 1（バランス関数）** $n$ ビットの入力変数に対して，1 ビットの出力を持った関数 $f\colon \{0,1\}^n \to \{0,1\}$ について，半分の入力パターンに対して 0，もう半分

1: 日大理工・院 (前)・数学　2: 日大理工・教員・数学

に対しては 1 を返す関数を **バランス関数** と定める．

今回の実験で用いるパリティ関数はバランス関数である．パリティ関数の場合，観測値の理論値は「11...1」のみとなる．Deutsch-Jozsa アルゴリズムは入力変数の桁数 $n$ に依存せず，1 度だけ関数 $f$ を実行すればどちらの関数であるかを判定できる．古典的な計算機でこれを判定するには，最悪の場合 $2^{n-1}+1$ 回の入力を試す必要があり，桁数 $n$ が大きくなるほど，実行回数の差は大きく開いていく．具体的な流れは次の通りである．[6]

1. 入力 $n$ ビットを 0，補助ビットを 1 で初期化．
2. すべての量子ビットに H ゲートを作用させる．
3. すべての量子ビットを関数 $f$ に入力する．
4. すべての量子ビットに H ゲートを作用させる．
5. 出力結果のうち補助量子ビット以外をすべて観測し，すべて 0 であれば関数 $f$ は定数関数，それ以外であればバランス関数と判定する．

## 4. 実験結果

ゲート数と深さはオラクル内のもののみ記載する．量子ビット数が 9 の場合の結果は表 1 に，量子ビット数が 25 の場合の結果は表 2 に記す．

表 1: 9 量子ビットの結果

| グラフ構造 | 辺数 | ゲート数 | 深さ | 評価値 |
|---|---|---|---|---|
| 完全グラフ | 36 | 8 | 8 | 0.2544 |
| トーラス | 18 | 10 | 10 | 0.2811 |
| 格子 | 12 | 10 | 10 | 0.2722 |
| 二分木 | 8 | 11 | 11 | 0.3123 |

表 2: 25 量子ビットの結果

| グラフ構造 | 辺数 | ゲート数 | 深さ | 評価値 |
|---|---|---|---|---|
| 完全グラフ | 300 | 24 | 24 | 0.5572 |
| トーラス | 50 | 40 | 35 | 0.5349 |
| 格子 | 40 | 38 | 37 | 0.5676 |
| 二分木 | 24 | 44 | 37 | 0.5539 |

## 5. 考察と今後の展望

今回の実験設定において，9 量子ビットの結果からはゲート数や深さ増大に伴い，評価値が大きくなる傾向が見られた．一方で，25 量子ビットの結果からは評価値において大きな違いは見られなかった．これは，ゲート数や深さの増大に伴いノイズの影響は増えたものの，理論値が 1 通りであることと，量子ビットの増加に伴い出現するビットパターンが指数関数的に増加したことで，$L^2$ ノルムが変動しにくくなったことに起因したものと考えられる．また，それぞれのグラフ構造において観測結果が理論値と一致した割合は，表の順に 45.4%, 47.8%, 44.5%, 45.7% となっている．今後の展望としては，深さが最小となる最適化手法を用いた追加実験による比較が挙げられる．現時点では格子についてのみ結果が得られており，その評価値は 0.5235 である．この手法を他のグラフ構造に適用できるように拡張し，評価値の比較を行うことが喫緊の課題である．加えて，他の量子アルゴリズムに対しても同様の実験的検証を行うことで，量子計算におけるノイズ耐性や効率性の観点から汎用的に適したグラフ構造を提案したい．

## 6. 参考文献


[1]嶋田義皓 著 and 情報処理学会出版委員会 監修. 量子コンピューティング = Quantum Computing : 基本アルゴリズムから量子機械学習まで. オーム社, 2020.11.

[2]Subhamoy Maitra and Partha Mukhopadhyay. “The Deutsch–Jozsa algorithm revisited in the domain of cryptographically significant Boolean functions”. International Journal of Quantum Information 3.02 (2005), pp. 359–370.

[3]Koji Nagata et al. “Quantum cryptography based on the Deutsch-Jozsa algorithm”. International Journal of Theoretical Physics 56.9 (2017), pp. 2887–2897.

[4]Ryuji Takagi et al. “Universal Sampling Lower Bounds for Quantum Error Mitigation”. Phys. Rev. Lett. 131 (21 Nov. 2023), p. 210602.

[5]IBM. The IBM Quantum heavy hex lattice. 2021. URL: `https://www.ibm.com/quantum/blog/heavy-hex-lattice`.

[6]中山茂 著. 量子アルゴリズム. 技報堂出版, 2014.1.