

強化学習を用いた ZX-Calculus による量子回路最適化手法の開発 —強化学習 × ZX-Calculus で切り開く量子回路最適化—

1. 背景

量子コンピュータは従来のコンピュータを超える計算能力を持つ可能性があり、金融、材料科学、機械学習などの分野での応用が期待されている。しかし、現在のハードウェア技術にはさまざまな課題があり、実用化に向けた大規模な計算を行うことが難しいのが現状である。

その中でも特に大きな問題として、量子ビット数の制約や、量子ゲート操作に伴う誤差、デコヒーレンスの影響等が挙げられる。量子コンピュータは本質的にノイズに弱いため、計算結果の信頼性を確保するためには、できるだけ少ないゲート数で計算を実行することが求められる。また、ハードウェアのリソースや性能が限られている現状では、計算コストを抑えながら効率的に計算を行う手法の開発が不可欠となっている。こうした背景から、量子回路最適化技術が重要視されており、その中でも ZX-Calculus による量子回路最適化手法が注目を集めている。[1] [2]

ZX-Calculus による量子回路最適化は、まず量子回路を ZX-Diagram と呼ばれるグラフに変換し、ZX-Diagram 上で簡約化操作を適用することで最適化し、最終的に量子回路を抽出することで、量子回路最適化を実現する。(図 1)

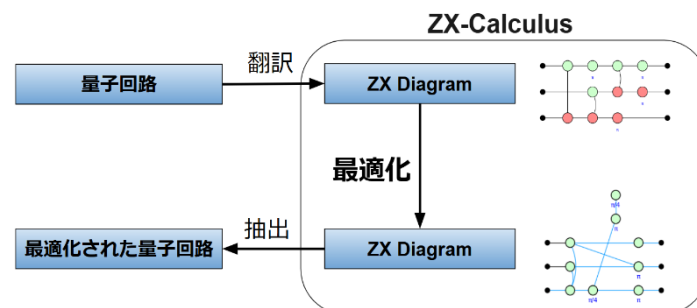


図 1. ZX-Calculus による量子回路最適化の概要

この手法の特徴は、量子回路ベースの最適化とは異なり、ZX-Diagram 上での簡約化操作を通じて、量子回路の大域的構造そのものを再構成できる点にある。そのため、量子回路ベースの最適化では得られない最適化結果が得られる可能性があり、大変有用な手法と考えられている。

しかしながら、これまで提案されている ZX-Calculus に基づく最適化手法には依然として課題が残されている。特に問題となるのは、「簡約化操作の順番が固定的であること」および「簡約化の終了条件が単純すぎること」である。従来手法では、いくつかの簡約化操作を決まった順序で適用するため、回路によっては最適な変換ができず、ゲート数がほとんど減らない場合がある。また、最適化の終了条件として「適用可能な簡約化操作がなくなった時点で終了する」というシンプルな基準を採用しているが、この基準では、不要な操作を加えかねず、実際に操作の前後で量子回路上のゲー

ト数が増加してしまうこともある。こうした問題が解決されなければ、ZX-Calculus による最適化の有効性は限定的なものになってしまう。

2. 目的

前述のような背景を踏まえ、本プロジェクトの目的は、強化学習により ZX-Diagram 上の簡約化操作の適用順序を学習させ、適用する簡約化操作の動的最適化を行うことで、ZX-Calculus による量子回路最適化手法の更なる性能向上を図るというものである。

3. ソフトウェア開発内容

本プロジェクトでは、以下のソフトウェアを作成した。

1) CZX

: ZX-Calculus 上の処理を高速化する ZX-Calculus の C++ ライブラリー

2) YN-Optimizer

: ZX-Diagram 上の簡約化操作の最適化を行う量子回路最適化ソフトウェア

a. YN-RL : 簡約化操作の最適化のための強化学習手法

b. YN-MCYS : 訓練済み YN-RL エージェントによるガイドをつけた
モンテカルロ木探索 (MCTS) による最適化手法

1) CZX

強化学習の開発中に、ZX-Diagram から量子回路に抽出する処理等に多大な時間を要していることが判明した。これらの処理は強化学習中に何度も使用され、学習時間への影響が大きい。このような課題を解決するために、探索および強化学習の高速化を図るため、C++ により実装した ZX-Calculus ライブラリー「CZX」を開発した。

CZX の実装は Python で実装された既存の ZX-Calculus ライブラリー PyZX [3] を基に行い、ZX-Diagram の特徴を考慮したデータ構造を採用し、C++ のムーブセマンティクスを活用し不要なコピーの回避による効率化を行った。これにより、ZX-Diagram から量子回路に戻す「抽出操作」、「抽出後の後処理」、ZX-Calculus の規則を適用する「簡約化操作」のいずれについても PyZX と比べて 10 倍近い高速化を達成した。(図 2)

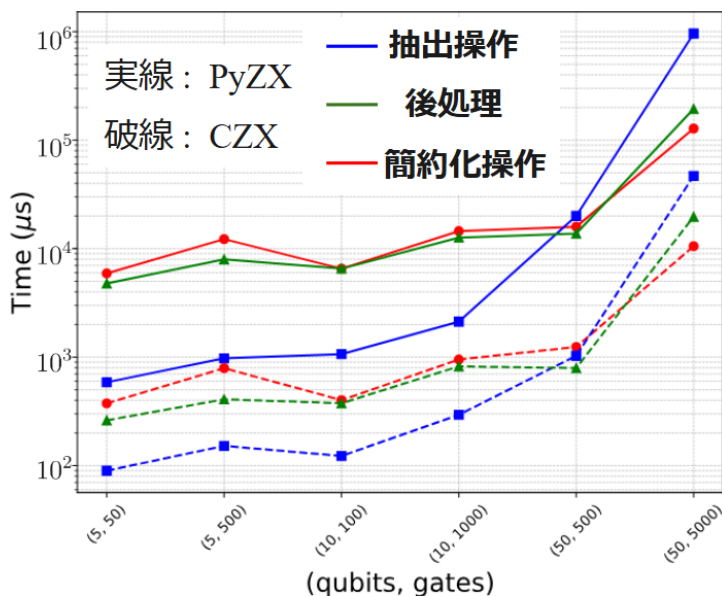


図 2. PyZX と CZX の速度比較

2) YN-Optimizer

a. YN-RL

ZX-Diagram 上の簡約化操作の適用順序を学習する強化学習手法として、「YN-RL」を開発した。YN-RL の強化学習のアーキテクチャの概要を説明すると、学習アルゴリズムとして Phasic Policy Gradient (PPG) [4] を採用し、Actor/Critic のネットワークとして、ZX-Diagram のグラフ構造をよく捉えられるように、Graph Neural Network (GNN)、具体的には GATv2 レイヤ [5] を採用した。

b. YN-MCTS

訓練済み YN-RL エージェントによるガイドをつけたモンテカルロ木探索手法として、「YN-MCTS」を開発した。YN-MCTS の概要としては、訓練済み YN-RL エージェントにより、探索不要なノードの枝刈りを行いつつ、有望なパスを MCTS により探索しながら発見していくというものであり、アイデアとしては、囲碁 AI として有名な Alpha Go Zero [6]に近い。

YN-RL, YN-MCTS の開発により、従来までの ZX-Calculus による量子回路最適化手法（従来手法）と比較して、最適化の性能を大幅に更新することができた。（図 3, 4）

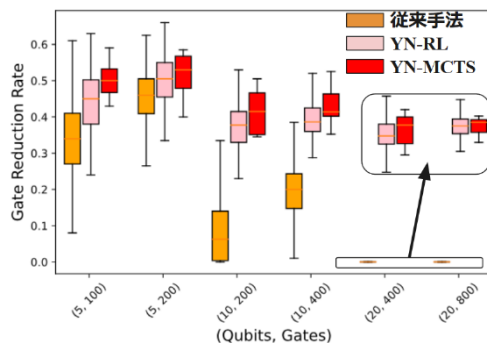


図 3. ランダム回路でのゲート削減率の比較

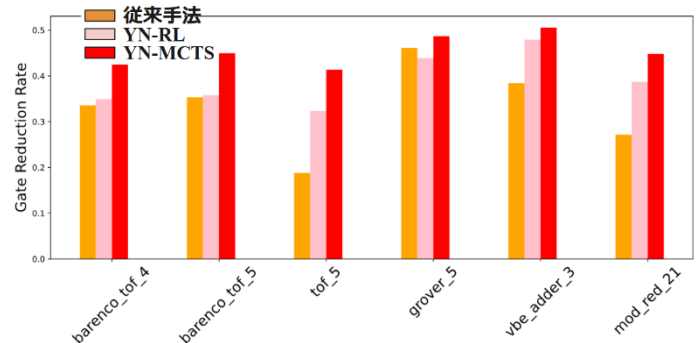


図 4. 小規模回路でのゲート削減率の比較

更に、YN-RL, YN-MCTS は ZX-Calculus ベース以外の最適化手法と比較しても、優れた最適化を実現し、我々の知る限り量子回路上のゲート削減のベストアルゴリズムである手法 [7] にも性能として肉薄している。（図 5）

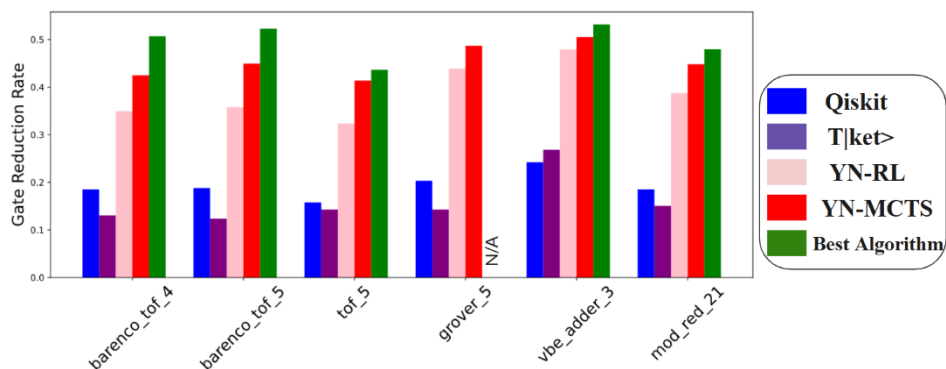


図 5. 既存手法との小規模回路でのゲート削減率の比較

4. 新規性・優位性

我々が開発した YN-Optimizer は、ZX-Diagram 上の簡約化操作の最適化に強化学習および、強化学習によるガイドをつけたモンテカルロ木探索などの、新規的な手法で実装され、従来までの ZX-Calculus による量子回路最適化手法と比較して、大幅に優れた最適化性能を達成し(図 3, 4)、ZX-Calculus による量子回路最適化手法の有用性・可能性を提示した。さらに、ZX-Calculus ベース以外の量子回路最適化手法と比較しても、YN-Optimizer は優れた最適化を実現しており、更に既存のベストアルゴリズムに肉薄する性能を発揮することができており(図 5)、優位性も十分である。

5. 期待されるユーザー価値と社会へのインパクト

本プロジェクトの成果は、ZX-Calculus による量子回路最適化手法の有用性と可能性を明確に示すものであり、今後の研究開発の指針となるものである。例えば、簡約化操作の最適化手法の更なる改善や、ZX-Diagram 上の簡約化操作の拡張、ZX-Calculus による最適化手法とその他の最適化手法を組み合わせたハイブリッド最適化等の実現により、ZX-Calculus による量子回路最適化手法の更なる性能向上が期待される。

このような量子回路最適化技術の性能向上は、量子コンピューティングの実用化にむけ大きな鍵となる。前述のようにノイズの影響を受けやすい現行の量子コンピュータにおいては、より短い回路長で等価な演算を実行できることが、計算の精度向上、実行可能性の向上や、計算コストの削減に直結する。更に、将来的に様々な量子ソフトウェアや量子アルゴリズムを実行する際にも、量子回路を圧縮することで、より効率的な計算の実行が可能になる。そのため、本プロジェクトの成果は、量子コンピューティングの基盤技術である量子回路最適化技術の発展に寄与し、延いては量子コンピューティングの実用化に向けた一助となる。

(参考文献)

- [1] <https://arxiv.org/abs/1902.03178>
- [2] <https://arxiv.org/abs/1903.10477>
- [3] <https://pyzx.readthedocs.io/en/latest/>
- [4] <https://arxiv.org/abs/2009.04416>
- [5] <https://arxiv.org/abs/2105.14491>
- [6] <https://www.nature.com/articles/nature24270>
- [7] <https://www.nature.com/articles/s41534-018-0072-4>

6. 氏名 (所属)

野上 大輝 (九州大学 理学部物理学科)

吉岡 真陸 (九州大学 理学部物理学科)