ゲート型量子計算機に実装した QAOA における確率的勾配降下法による変分パラメータの最適化

Optimization of QAOA with Quantum Processing Unit via Stochastic Gradient Descent

東京農工大 ○沖田涼、三木司、島田萌絵、白樫淳一

Tokyo University of Agriculture & Technology

°R. Okita, T. Miki, M. Shimada, and J. Shirakashi

E-mail: s213856z@st.go.tuat.ac.jp

現在のゲート型量子計算機は、誤り訂正を行わない Noisy Intermediate-Scale Quantum device (NISQ)と呼ばれている。NISQ では大規模な回路を実行することが困難であるため、小規模な Quantum Processing Unit (QPU)での量子状態の生成と古典計算機(CPU)での計算処理を組み合わせた量子・古典ハイブリッドアルゴリズムが注目されている。その 1 つとして、変分原理と量子断熱計算に基づき組合せ最適化問題の解探索を行う、Quantum Approximate Optimization Algorithm (QAOA)[1]が提案されている。NISQ で生成した量子状態は緩和時間やゲートエラーの影響を顕著に受けるため、QPU の実行回数を始めとする計算コストを抑えた QAOA の実装方法が求められている。これまで我々は、NISQ を用いて実装した QAOA の演算特性について検討してきた[2]。今回は、QAOA 特有の異なる問題が共通して持つ解空間の特徴[3]を効率的に探索可能であると考えられる確率的勾配降下法(Stochastic Gradient Descent: SGD)[4]および、ノイズに強く高速に多変数関数の勾配計算が可能な Simultaneous Perturbation Stochastic Approximation (SPSA)[5]を組み合わせ、変分パラメータの最適化における QPU の実行回数を低減させることを検討した。

本研究では、IBM Quantum[6]からアクセスできる IBM Q を QPU として用いた。図 1 は SGD を用いた QAOA アルゴリズムの概要である。まず、QPU 上でn個の学習データからランダムに抽出したグラフについてp層の QAOA 回路を実行し量子状態$|\psi(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle$を生成する。問題のハミルトニアン$H_C$について期待値の勾配を計算し、期待値が減少する方向へ学習率ηの値に応じて変分パラメータ$(\boldsymbol{\gamma},\boldsymbol{\beta})$を更新する。この処理を学習データ全てについて行った後、その時点での変分パラメータを用いてテストデータm個に対し回路を実行し解を得た。今回は、学習データおよびテストデータとして 5 ノードのランダムグラフを 100 インスタンスずつ独立に生成し、それらに対する Max-Cut 問題をH_Cとしてマッピングした。1 層の回路による実験を行った結果、SPSA を用いて勾配計算を行った場合、厳密解に対する平均の近似比は 0.966、1 インスタンス当りの QPU の実行回数は 2 回となった。以上の結果から SGD を用いた変分パラメータの最適化により、SGD を用いなかった場合[2]と同程度の精度を保ちながら QPU の実行回数を削減できることが示唆された。

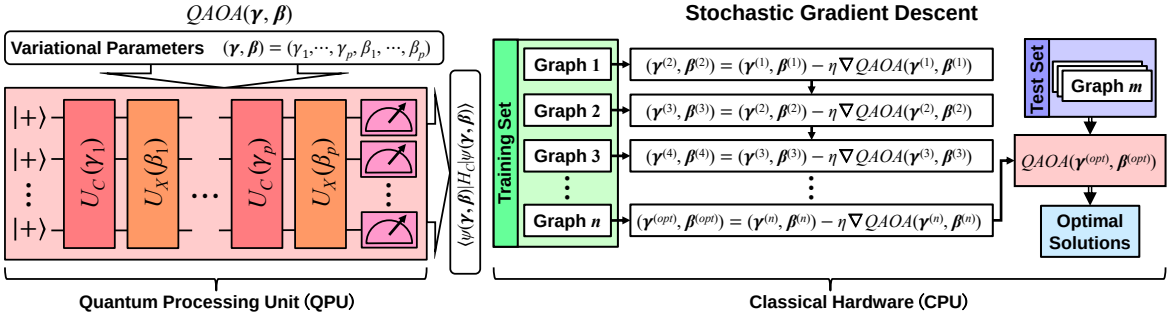

Fig. 1 Implementation of quantum approximate optimization algorithm with stochastic gradient descent.

References

[1] E. Farhi, J. Goldstone, and S. Gutmann, arXiv:1411.4028v1 [quant-ph] (2014).

[2] 沖田、三木、島田、白樫: 第 68 回応用物理学会春季学術講演会 19a-Z34-9 (2021).

[3] F. G. S. L. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, arXiv:1812.04170v1 [quant-ph] (2018).

[4] H. Robbins, and S. Monro, Ann. Math. Statist. 22 (1951) 400.

[5] J. C. Spall, Johns Hopkins APL Tech. Dig. 19 (1998) 482.

[6] IBM Quantum,“https://quantum-computing.ibm.com/” (2021).