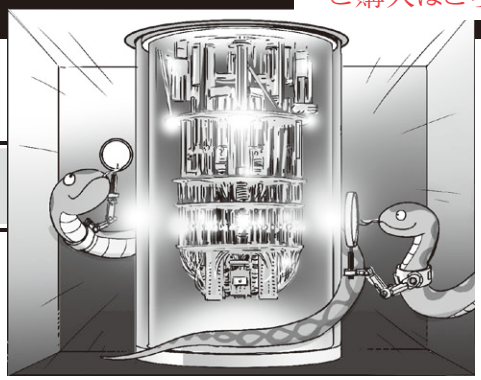
ご購入はこちら

古典コンピュータと量子コンピュータを併用する
量子古典ハイブリッド・アルゴリズムの使用例

Pythonで体験！ 量子コンピュータ



後編 ナップサック問題や機械学習などの実問題に挑戦

辻野 孝一

現在（2022年4月）の量子コンピュータは，演算中に発生するエラーが原因で正しい計算を行うことが困難です．このエラーを訂正する機能が実現されて初めて量子コンピュータは実用的な計算を行うことができると言われています．しかし誤り訂正を備えた「誤り耐性量子コンピュータ」の実現はまだ先のことだと言われています．

現在ではエラー訂正ができない小/中規模（数百量子ビット程度）の量子コンピュータの利用方法の開発が進められています．このような量子コンピュータをNISQ（Noisy Intermediate-Scale Quantum）デバイスと呼びます．NISQでは古典コンピュータと量子コンピュータを併用する量子古典ハイブリット・アルゴリズムというアプローチが主流です．このようなアプローチで量子化学計算や量子機械学習，最適化，金融などへの応用が進められています．

本稿では，Pythonでプログラミングを行うことで量子古典ハイブリット・アルゴリズムを体験します．今回は量子古典ハイブリッド・アルゴリズムの1つであるQAOA（Quantum Approximate Optimization Algorithm，量子近似最適化アルゴリズム）を用いてさまざまな実問題を解いていきます．実行環境はIBM Quantum LabのNotebook（Jupyter NotebookのようなPythonの実行環境）を利用します．詳細は前編（2022年7月号）を参照してください．本稿の目次は次の通りです．

1 ポートフォリオ最適化
2 ナップサック問題
3 蓄電池システムの収益最大化
4 量子古典ハイブリッド機械学習

（編集部）

● 量子古典ハイブリッド・アルゴリズムの1つであるQAOA

QAOA（Quantum Approximate Optimization Algorithm，量子近似最適化アルゴリズム）は，組み合わせ最適化問題を解くためのアルゴリズムです．QAOAは量子古典ハイブリッド・アルゴリズムの1つです．

組み合わせ最適化問題とは，さまざまな制約条件のもとで，数多くの選択肢の中から，目的となる指標（コスト関数）を最も小さくする変数の組み合わせを求める問題のことです．組み合わせ最適化問題には，巡回セールスマン問題や配車ルートの問題，集合被覆問題，ナップサック問題，スケジューリング問題などがあります．

これらの問題には，意味のある時間内で正確な解を見つけることが非現実的なものが含まれるため，近似アルゴリズムが必要となります．このアルゴリズムを理解するには，量子断熱計算と呼ばれる知識が必要ですが，ここではQAOAの使い方の説明をすることにとどめます．QAOAを実行する手順は次の通りです．

- ステップ1：問題の定式化
- ステップ2：コスト関数に変換
- ステップ3：量子回路の作成
- ステップ4：量子回路と古典オプティマイザを用いて最適なパラメータを求める
- ステップ5：得られたパラメータでQAOA回路を作り結果を分析

上記のステップ1～ステップ5の手順を量子ゲートのレベルから実装するためには，複雑な手続きが必要になります．ここでは，QAOAの量子ゲート回路などを，プログラムすることなく問題を解くことができる高レベルAPIを使って解説を行います．

1 ポートフォリオ最適化

ポートフォリオ最適化をQAOAを用いて実行する方法を解説します．

● 問題設定

ポートフォリオとは，株式などの金融資産の組み合わせのことです．ポートフォリオ最適化のゴールは，リスクを最小化し，リターンを最大化することです．リスクとリターンは通常，トレードオフの関係にあります．積極的に大きなリターンを得ようとするなら，リスクが大きい金融資産を多く所有し，手堅く運用し