

2025/05/14
理論談話会#4

Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm

Azad, U., Behera, B. K., Ahmed, E. A., Panigrahi, P. K., & Farouk, A. (2022). Solving vehicle routing problem using quantum approximate optimization algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 24(7), 7564-7573.

M1 村山拓未

概要

- Vehicle Routing Problem (VRP) をQuantum Approximate Optimization Algorithm (QAOA) で解く可能性を検討
- VRPをイジングモデルに変換し、QAOAで最適化 (Qiskitを用いて実装)
- 提案手法の性能を、古典的最適化ソルバーCPLEXと比較し、最大15量子ビットの小規模問題で評価をする
- QAOAの性能は古典的最適化アルゴリズム、回路深さ p 、初期パラメータおよび問題インスタンスによって大きく左右される

新規性・有用性・信頼性

新規性

- 一般的なVRPに対するハイブリット量子-古典アルゴリズム（QAOA）の適用を検討
- VRPのQUBO・Ising形式による定式化を提示

有用性

- 計算が難しいNP困難な組み合わせ最適化問題に対するアプローチ
- 量子計算が古典計算よりも計算上の有意性を持つ可能性

信頼性

- 詳細なQUBO・Ising形式による理論的定式化と複数の小規模インスタンスでのIBM Qiskitを用いたシミュレーションによりQAOAの性能を評価
- 既存の古典ソルバーCPLEXとの比較分析

I. Introduction

Introduction

高度道路交通システム（ITS）

最先端の情報通信技術を用いて人と道路と車両とを情報でネットワーク

→よりスマートな意思決定による交通サービスの効率向上

ITS技術が使われている例：物流

サプライチェーン管理とスケジューリング

- 最適経路を選択することで燃料消費の削減、時間短縮
- 車両交通の管理を効果的に行い渋滞緩和

ITSの課題

→計算困難な最適化問題

→既知の最適化問題へのマッピング、厳密解法、近似解法を用いる

Introduction

ITS技術の応用例：VRP

- NP困難な組み合わせ最適化問題
- 各地点を一度だけ訪問する最適な配送経路を考える

既存のVRPの解法

- 厳密アルゴリズム：分枝限定法（大規模インスタンスで指数関数的時間）
- ヒューリスティックアルゴリズム：効率的な探索空間探索
- 二段階アルゴリズム：分割統治法（小規模インスタンスに分割）
- メタヒューリスティックアルゴリズム：遺伝的アルゴリズム、シミュレーテッドアニーリング

Introduction

近年の量子コンピューティング技術の発達

- 古典的解法に対する計算上のアドバンテージ
- NISQ（ノイズあり中規模量子デバイス）の限界

新たな解法：QAOA（Quantum Approximate Optimization Algorithm）

- Hybrid Quantum-classical (HQC) アルゴリズムの一種
- 計算上のボトルネックのみに量子デバイスを適用し、深い量子回路を必要としない

本研究ではQAOAをVRPに適用

→そのためにVRPのQUBO形式をIsing形式に変換する

II. Vehicle Routing Problem

Vehicle Routing Problem

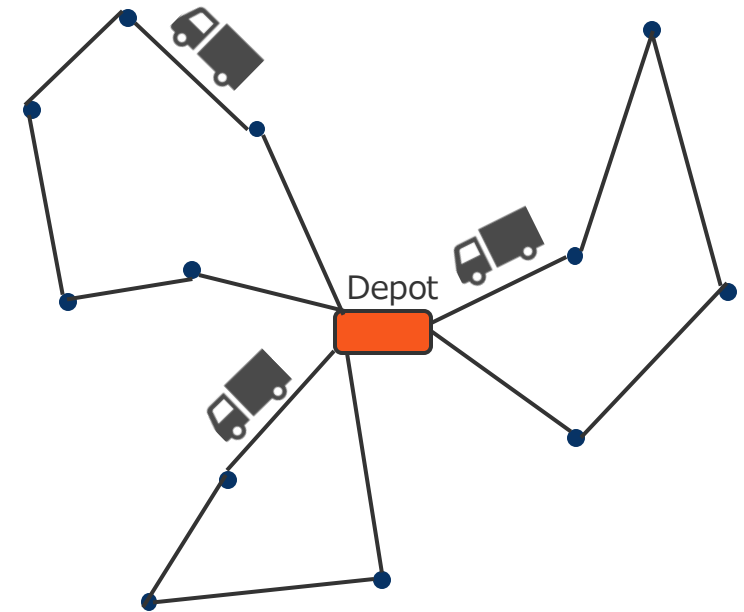
VRPの基本設定

- NP困難な組み合わせ最適化問題
- 巡回セールスマン問題の一般化
- インスタンス (n, k)
 - k 台の車両と $n - 1$ 個の地点（残り1つはデポ D ）
- 全車両がデポから出発しデポに戻り、各地点は必ず一度訪問される時の総移動距離を最小化

現実世界のVRP

→様々な制約がある（車両容量、時間制約等）

ここでは制約なしの基本的なVRPについて考える



Energy Functional for VRP

定式化

$$VRP(n, k) = \min_{\{x_{ij}\}_{i \rightarrow j} \in \{0,1\}} \sum_{i \rightarrow j} w_{ij} x_{ij}$$

→総移動距離の最小化

x_{ij} : $n \times (n - 1)$ 個 の決定変数。 $i \rightarrow j$ のエッジの有無をバイナリーに表す。

w_{ij} : $i \rightarrow j$ の距離を表す重み。

制約

$$\left. \begin{array}{l} \sum_{j \in \text{source}[i]} x_{ij} = 1 \quad \forall i \in \{1, \dots, n-1\} \\ \sum_{j \in \text{target}[i]} x_{ij} = 1 \quad \forall i \in \{1, \dots, n-1\} \end{array} \right\} \text{全てのノードは一度だけ訪問される}$$

$\text{source}[i]$: i から出ていくノード先の集合

$\text{target}[i]$: i に入ってくるノード元の集合

$$\left. \begin{array}{l} \sum_{j \in \text{source}[0]} x_{0j} = k \\ \sum_{j \in \text{target}[0]} x_{j0} = k \end{array} \right\} \text{全ての車両はデポから出発しデポに帰ってくる}$$

Energy Functional for VRP

目的関数と制約を用いたエネルギー汎関数 H_{VRP} を定式化

$$H_{VRP} = H_A + H_B + H_C + H_D + H_D$$

$$H_A = \sum_{i \rightarrow j} w_{ij} x_{ij}$$

$$H_B = A \sum_{i \in 1, \dots, n-1} \left(1 - \sum_{j \in \text{source}[i]} x_{ij} \right)^2$$

$$H_C = A \sum_{i \in 1, \dots, n-1} \left(1 - \sum_{j \in \text{target}[i]} x_{ij} \right)^2$$

ただし A は与えられた問題インスタンスにおける最大の重み w_{ij} に比例する任意の正の定数

$$H_D = A \left(k - \sum_{j \in \text{source}[0]} x_{0j} \right)^2$$

$$H_E = A \left(k - \sum_{j \in \text{target}[0]} x_{j0} \right)^2$$

QUBO Formulation of VRP

制約付きの最適化問題（VRP）を二次制約なし二値変数最小化問題（QUBO）に変換

変数ベクトルの定義

すべての決定変数 x_{ij} を1本の列ベクトルにまとめる

$$\vec{x} = [x_{(0,1)}, x_{(0,2)}, \dots, x_{(1,0)}, x_{(1,1)}, \dots, x_{(n-1,n-2)}]^T$$

制約式をベクトル内積で表現

$$\sum_{j \in \text{source}[i]} x_{ij} = \vec{z}_S^T[i] \vec{x} : i \text{ から出るエッジ}$$

$$\sum_{j \in \text{target}[i]} x_{ji} = \vec{z}_T^T[i] \vec{x} : i \text{ に入るエッジ}$$

$\vec{z}_S^T[i]$: i から出るエッジのみ1、それ以外は0のマスクベクトル

$\vec{z}_T^T[i]$: i に入るエッジのみ1、それ以外は0のマスクベクトル

QUBO Formulation of VRP

ベクトル内積で表現した制約をエネルギー汎関数に代入し**2次形式** ($\vec{x}^T Q \vec{x} + \vec{g}^T \vec{x} + c$) に整理

$$\begin{aligned} H_{VRP} = & A \sum_{i=0}^{n-1} (\vec{z}_{S[i]} \vec{z}_{S[i]}^T + \vec{z}_{T[i]} \vec{z}_{T[i]}^T) \vec{x}^2 \\ & + \vec{w}^T \vec{x} - 2A \sum_{i=1}^{n-1} (\vec{z}_{S[i]}^T + \vec{z}_{T[i]}^T) \vec{x} \\ & - 2Ak(\vec{z}_{S[0]}^T + \vec{z}_{T[0]}^T) \vec{x} + 2A(n-1) + 2Ak^2 \end{aligned}$$

$$\begin{aligned} Q = & A([\vec{z}_{T[1]}, \vec{z}_{T[2]}, \dots, \vec{z}_{T[n-1]}, \vec{z}_{T[0]}, \vec{z}_{T[2]}, \dots, \vec{z}_{T[n-2]}]^T, \\ & + [[\vec{z}_{S[0]}]^{\times(n-1)} [\vec{z}_{S[1]}]^{\times(n-1)} \dots [\vec{z}_{S[n-1]}]^{\times(n-1)}]) \\ \vec{g} = & \vec{w} - 2A(\vec{J} + \vec{K}) - 2Ak(\vec{z}_{S[0]} + \vec{z}_{T[0]}) \\ c = & 2A(n-1) + 2Ak^2 \end{aligned}$$

※ $\vec{J}$ はデポからのエッジのみ 0
 $\vec{K}$ はデポへのエッジのみ 0

対応する係数 $Q(n(n-1) \times n(n-1))$, $\vec{g}(n(n-1) \times 1)$, c は以下の意味を持つ

- Q : ノードごとの訪問回数による制約に関係し、変数間の相互作用
- $\vec{g}$: エッジごとの距離 + デポ出発制約に関係する個別の変数にかかるコスト
- c : 最適化には影響しない定数項

Ising Formulation for VRP

バイナリ変数 $x_{ij} \in \{0, 1\}$

$$x_{ij} = \frac{s_{ij} + 1}{2}$$



スピン変数 $s_{ij} \in \{-1, 1\}$

2次形式におけるすべてのバイナリ変数をスピン変数に変換して整理

$$H_{Ising} = - \sum_i \sum_{j < i} I_{ij} s_i s_j + \sum_i h_i s_i + d$$

$$\begin{aligned} I_{ij} &= -\frac{Q_{ij}}{4} \quad \forall i < j, & I_{ii} &= 0 \quad \forall i \\ h_i &= \frac{g_i}{2} + \sum_j \frac{Q_{ij}}{4} + \sum_j \frac{Q_{ji}}{4} \\ d &= c + \sum_i \frac{g_i}{2} + \sum_i \frac{Q_{ii}}{4} + \sum_i \sum_j \frac{Q_{ij}}{4} \end{aligned}$$

スピン s_i をパウリ演算子 σ_i^z に置き換えることで量子回路上で表現できるハミルトニアンとなる
位相を反転させる量子演算子の1つ

Vehicle Routing Problem

VRPの種類（制約つき）

- 時間枠付きVRP（VRPTW）：特定の時間枠内に訪問を完了する
- 容量制約付きVRP（CVRP）：車両の容量が非負かつ総容量未満

VRPに用いられる最適化

- 古典的最適化手法
 - 厳密解法
 - ヒューリスティックス
 - メタヒューリスティックス
- 量子ベースの最適化手法
 - 量子アニーリング
 - 量子ウォーク



現状の量子デバイスの限界

→量子と古典のハイブリット手法であるQAOAを用いる

必要な事前知識

量子コンピュータについてGPTに聞いてみた

	量子コンピュータ（QAOAベース）	古典コンピュータ（例：CPLEXなど）
情報の基本単位	量子ビット（qubit）状態は 0 と 1 の重ね合わせ	ビット（bit）0 または 1 のいずれか
並列性	量子並列性 あり：多数の経路（状態）を同時に持てる	並列性は マルチスレッド や GPU など基本は逐次的処理
探索方法	状態空間を重ね合わせ状態で同時に探索（干渉・測定で解を導く）	枝刈り、線形緩和、ヒューリスティックなどの 逐次探索
制約処理	制約をペナルティ項としてハミルトニアンに組み込む	制約を明示的に扱える（線形計画、整数制約）
主なアルゴリズム	QAOA（量子近似最適化）量子断熱、Groverなど	線形/整数計画、動的計画法、メタヒューリスティクス（GA, SA）
最適化精度	近似解。最適性保証はないが、パラメータ次第で高精度に近づく	精密な最適解（特に小～中規模）保証付きアルゴリズムもあり
計算コスト（理論）	一部の問題で 指数時間 → 多項式時間の可能性 あり	NP困難問題に対しては指数的に時間が増大
現実的制約	ハードウェアは まだ小規模（数10～数100量子ビット） ノイズやエラーが多い	商用環境で大規模な問題も処理可能（メモリ・並列化済）
ハイブリッド性	古典最適化器と組み合わせて最適なパラメータを探す（VQA構造）	単体で完結（ただし量子ソルバーと連携可能なAPIも）
代表的な用途（本論文文脈）	VRP、TSP、MaxCutなどの組合せ最適化問題の 新たな解法の可能性	VRPなどは長年の改良により 実用的な解法が確立 されている

量子ビットの基礎

古典コンピュータ：情報は 0,1 で表現



量子ビット：異なる2つの状態の重ね合わせを許す

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad \alpha, \beta \text{ は複素数 (複素確立振幅)}$$

複素確率振幅

- 測定という操作をして初めて0か1かが確率的に決まる
- 測定結果が0になる確率 $p_0 = |\alpha|^2$
- 測定結果が1になる確率 $p_1 = |\beta|^2$
- $|\alpha|^2 + |\beta|^2 = 1$

ブラケット記法

- $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, 列ベクトル → ケット
- $\langle\psi| = (\alpha^* \quad \beta^*)$, 行ベクトル → ブラ
- 異なるブラとケットを抱き合わせる → 内積
- 異なるブラとケットを背中合わせ → 演算子 (行列)

例) 0と1が同じ重みで重ね合わさった状態

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

また、複素数であるため

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{i\phi}|1\rangle)$$

も存在し、この時 α と β に偏角 (位相と呼ぶ)

量子ビットに対する演算

量子ビットに対する演算の特徴

- 線形性：量子コンピュータ上での操作は状態ベクトルに対する線形変換、 2×2 の複素行列で表現
- ユニタリ性：量子ビットに対する操作はユニタリ行列で表現

量子ゲートの種類

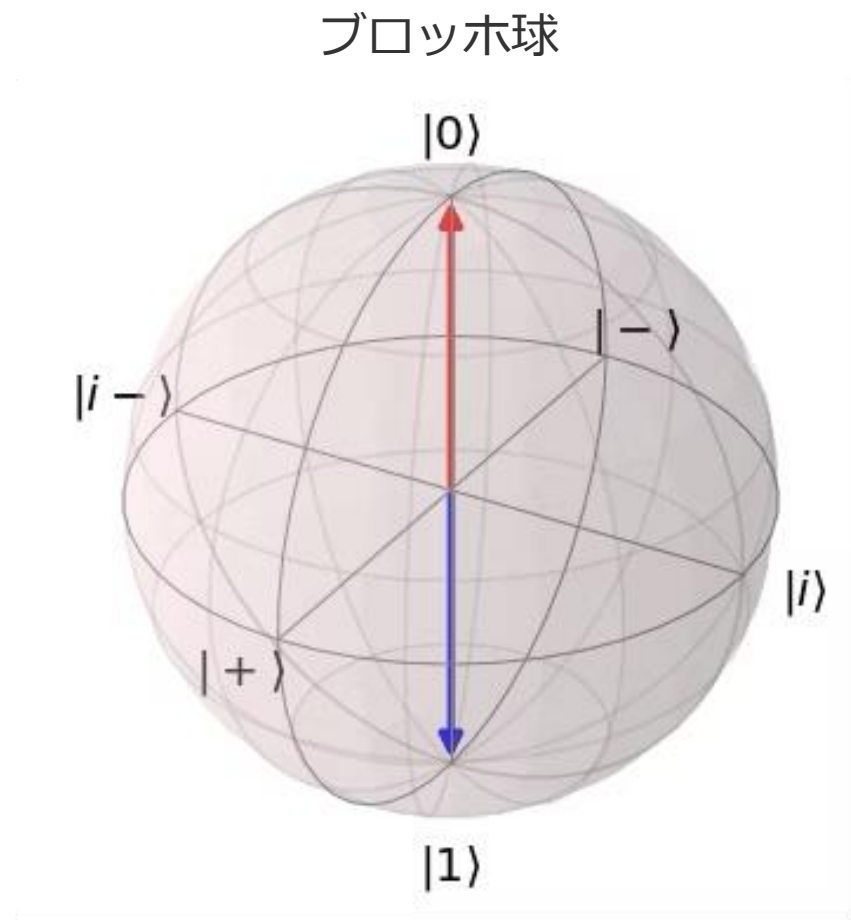
- パウリゲート：X、Y、Zゲートの3種で各軸周りを180度回転する操作
- 回転ゲート：特定の軸を中心として量子ビットの状態を回転させる操作、角度 θ によってパラメータ化される
- 制御ゲート：制御量子ビットが特定の状態（通常は $|1\rangle$ ）にある時のみ、ターゲット量子ビットに対して実行
 - 例：CNOT（制御Xゲート）

具体例

- 恒等演算子： $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1|$
- パウリX演算子： $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |1\rangle\langle 0| + |0\rangle\langle 1|$
 - ビットの反転、例えば $X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$
- パウリZ演算子： $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|$
 - 位相を反転、例えば $Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$

参考：ブロッホ球

1量子ビットが取りうる状態を幾何的に表す3次元の球



III. Quantum Approximate Optimization Algorithm

Adiabatic Quantum Computation (AQC)

断熱量子計算 (AQC)

- 困難な組み合わせ最適化問題を解くために最初に用いられた量子計算モデル
- Gate-based quantum computingとは異なり、量子力学の**断熱定理**に基づく
- 2つのハミルトニアン、 $\hat{H}_{mixer}$ と $\hat{H}_{cost}$ が計算に必要
 - $\hat{H}_{mixer}$ は容易に準備可能な初期状態を作るためのハミルトニアン (通常は $|+\rangle^{\otimes N}$)
 - $\hat{H}_{cost}$ は最適化問題の目的関数をエンコード、基底状態が問題の最適解に対応する
 - ハミルトニアンは局所的：ごく少数の量子ビット間の相互作用のみで構成される
→ゲート数が少なくなり量子回路上で実装しやすい
 - 瞬間的なハミルトニアン： $\hat{H}(t) = (1 - t)\hat{H}_{mixer} + t\hat{H}_{cost}$
- AQCの計算時間はスペクトルギャップ ΔE (基底状態と第一励起状態のエネルギー差) に依存し、 ΔE が小さくなると指数関数的に計算時間が長くなる
 - 特定の困難な最適化問題インスタンスではこれが小さくなる傾向がある

Quantum Approximate Optimization Algorithm

QAOAの発想：断熱経路の離散化

- AQCにおけるスペクトルギャップによるステップサイズの制限をなくす
- AQCにおける理想的な断熱経路を有限のステップ数 p で粗く離散化して推測

量子状態の時間発展を表す演算子

QAOAのユニタリをステップ数 p で分解して近似（トロッター分解）

$$U = U(\hat{H}_{mixer}, \beta_0) U(\hat{H}_{cost}, \gamma_0) \dots (\hat{H}_{mixer}, \beta_{p-1}) U(\hat{H}_{cost}, \gamma_{p-1})$$

ハイブリッド量子-古典（HQC）アルゴリズム

- 量子コンピュータはパラメータ (β, γ) に基づく量子回路を実行し、最終状態の期待値を計算
- 古典コンピュータは量子回路で得られた測定結果の情報を用いてパラメータ (β, γ) をチューニング
- パラメータ (β, γ) は各ステップでの回転量を決めている

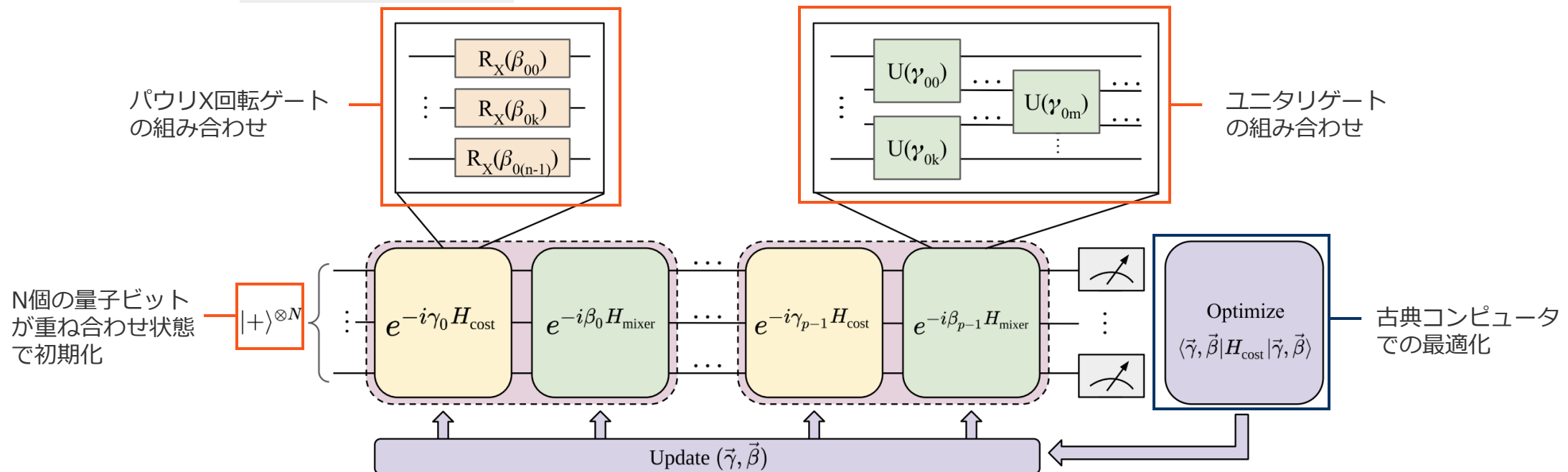
IV. Simulations and Results

Simulations

基本設定

- VRPの問題インスタンス： n 地点、 k 台の車両と距離行列 D
- IBM Qiskitを用いてQAOAを実行し、古典的な最適化ソルバーであるCPLEXと結果を比較
- 必要な量子ビット数： $N = n \times (n - 1) \rightarrow$ 各量子ビットが2つのノード間のエッジの可能性を表現

QAOAの回路図



ハミルトニアンと量子ゲート

ハミルトニアンの設定

- $\hat{H}_{mixer} = -\sum_{i=0}^{n \times (n-1)} \sigma_i^x$
 - 全ての量子ビットに対するパウリX演算子の総和
- $\hat{H}_{cost} = -\sum_i \sum_{j < i} l_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z - d$
 - VRPのIsing形式に基づく

l_{ij} : 量子ビット i, j 間の結合強度 (2変数の相互作用)
 h_i : 量子ビット i 単体のエネルギー寄与 (1次項)
 d : 定数項 (最適化に影響しない)

各Ising項に対応する量子ゲート

- $\sigma_i^z \sigma_j^z$: CNOT $\rightarrow$ RZ(θ) $\rightarrow$ CNOT q_i, q_j が共に1のとき q_i を回転
 - σ_i^z : RZ(θ) z 軸周りに q_i を回転 (位相を変える)
 - σ_i^x : RX(θ) x 軸周りに q_i を回転 (反転させる)
- 状態に位相をつける
- 状態を混ぜることで新しい選択枝を模索

量子回路と状態の進化

QAOA回路での状態進化

$$|\vec{\gamma}, \vec{\beta}\rangle = e^{-i\hat{H}_{mixer}\beta_p} e^{-i\hat{H}_{cost}\gamma_p} \dots e^{-i\hat{H}_{mixer}\beta_0} e^{-i\hat{H}_{cost}\beta_0} \underbrace{|+\rangle}_{|0\rangle \text{ と } |1\rangle \text{ が等確率で存在}}^{n \otimes (n-1)}$$

$\hat{H}_{mixer}$ の基底状態：全量子ビットが $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ の状態
|0>にHadamardゲートを作用させて実現

$e^{-i\hat{H}_{cost}\gamma}$ の役割：振幅は変化させずに、異なる計算基底状態の相対的な位相を調整し量子もつれを生成
 $e^{-i\hat{H}_{mixer}\beta}$ の役割：異なる計算基底状態の振幅を混合し状態の重ね合わせを強める

最終状態 $|\vec{\gamma}, \vec{\beta}\rangle$ を用いて期待値（平均エネルギー）を計算

$$E = \langle \vec{\gamma}, \vec{\beta} | \hat{H}_{cost} | \vec{\gamma}, \vec{\beta} \rangle$$



E を古典的なソルバーで最小化
最適パラメータ $\{\hat{\gamma}^*, \hat{\beta}^*\}$ を求める

最適なパラメータが得られるまでハイブリッド量子-古典アルゴリズムを繰り返す

VRPの最終的な解の取得

最適パラメータ $\{\hat{\gamma}^*, \hat{\beta}^*\}$ を用いて最終的な量子状態を計算する

※ $\hat{\gamma}^*, \hat{\beta}^*$ は量子の回転角を表す、QAOA回路で使うべき最適なパラメータ

QAOAで得られた最終的な量子状態は様々なビット列の重ね合わせ

→これを測定すると、状態数 $2^{n \times (n-1)}$ 個の1つのビット列に崩壊する（例：001101…）

→このビット列は隣接行列を平坦化したもの

最終的なコストは測定されたビット列において1となる（エッジが存在）距離を合計して計算

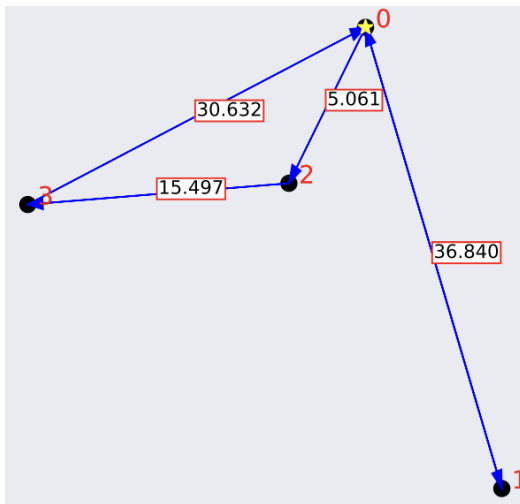
QAOA Experiments

Experiment 1

- インスタンス：(4, 2) → 量子ビット数 = $4 \times 3 = 12$ 個

- 距離行列 $D_1 = \begin{bmatrix} 0. & 36.84 & 5.06 & 30.63 \\ 36.84 & 0. & 24.55 & 63.22 \\ 5.06 & 24.55 & 0. & 15.50 \\ 30.63 & 63.22 & 15.50 & 0. \end{bmatrix}$

得られた最適経路

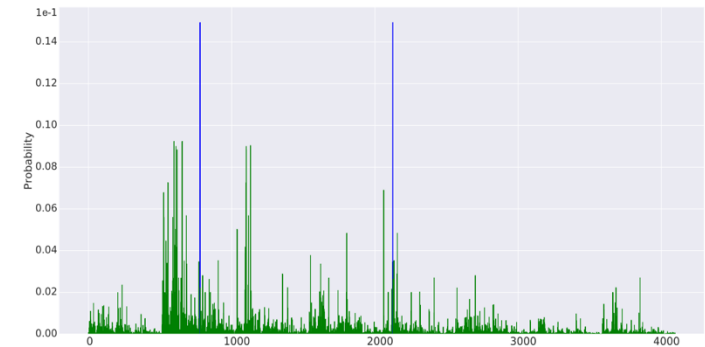


対応する隣接行列

$$A_1 = \begin{bmatrix} X & 1 & 1 & 0 \\ 1 & X & 0 & 0 \\ 0 & 0 & X & 1 \\ 1 & 0 & 0 & X \end{bmatrix}$$

総コスト

$$C_1 = 30.632 + 15.497 + 5.061 + 2 \times 36.840 = 124.871$$



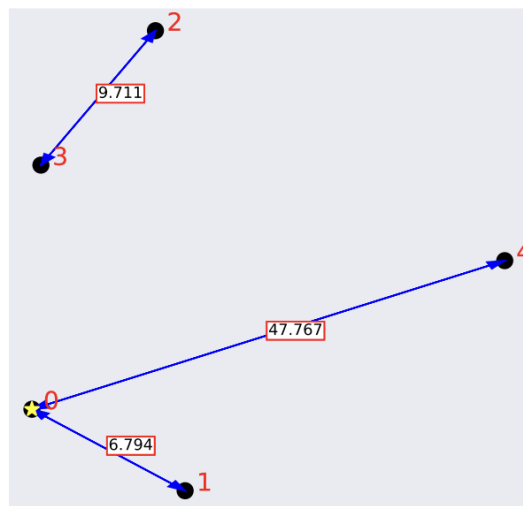
$p \geq 12$ の時正しい解（青線）が十分な確率で得られた
グラフは $p = 12$ のとき

QAOA Experiments

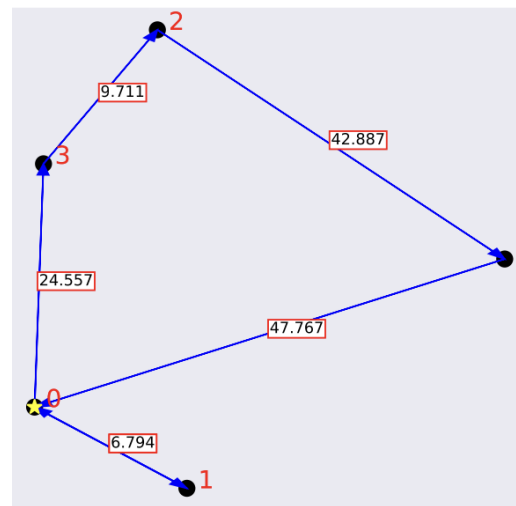
Experiment 2

- インスタンス：(5, 2) → 量子ビット数 = $5 \times 4 = 20$ 個
- 異なる p の値（6から40）でQAOAを実行

QAOAから得られた解の1つ



本来得られるべき最適解



コストは低いが制約を満たさない解が存在

→ 制約違反に対するペナルティが不十分

→ ペナルティ項を増大させることで回避できる可能性

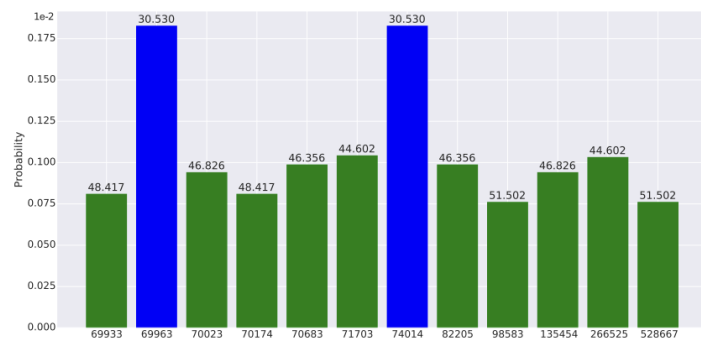
QAOA Experiments

Experiment 3

- インスタンス：(5, 3) → 量子ビット数 = 5 × 4 = 20個
- $p \geq 24$ で正しい解が十分な確率で得られた
 - 最適解はインデックス69963と74014に対応する状態

距離行列

$$D_4 = \begin{bmatrix} 0. & 12.138 & 0.32 & 7.2 & 2.626 \\ 12.138 & 0. & 16.307 & 5.3 & 17.021 \\ 0.32 & 16.307 & 0. & 9.309 & 2.98 \\ 7.2 & 5.3 & 9.309 & 0. & 16.759 \\ 2.626 & 17.021 & 2.98 & 16.759 & 0. \end{bmatrix}$$



$p = 24$ における上位12個の実現可能解

$$A_4 = \begin{bmatrix} X & 1 & 1 & 0 & 1 \\ 0 & X & 0 & 1 & 0 \\ 1 & 0 & X & 0 & 0 \\ 1 & 0 & 0 & X & 0 \\ 1 & 0 & 0 & 0 & X \end{bmatrix}$$

69963の隣接行列



比較的大きなインスタンスでもQAOAは有効
(QAOAの性能は量子回路深さ p に依存)

Comparative Analysis

VRPに対するQAOAの有用性を実証

→ 量子回路の深さ p や使用する古典的optimizerがQAOAの性能にどのように影響するか

分析の設定

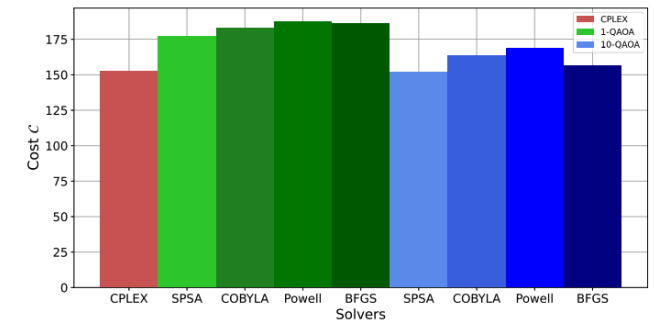
- 対象インスタンス：(4, 2)、(5, 2)、(5, 3) の3種類
 - ランダムにサンプリングされた距離行列 D によってそれぞれ150個の問題インスタンスを生成
- 比較対象：QAOAによる結果と古典的最適化ソルバーのCPLEXによる結果を比較
- 評価方法：各問題インスタンスについて得られた解のコストを平均して比較
 - QAOAでは5回試行した中で最も良い結果を採用
- QAOAの設定
 - 量子回路の深さを示すパラメータ p として、 $p = 1$ と $p = 10$ の場合を比較
 - QAOAの変分パラメータを最適化する古典的なoptimizerとしてSPSA、COBYLA、Powell、BFGSを使用（最大反復回数は5000回）

Comparative Analysis

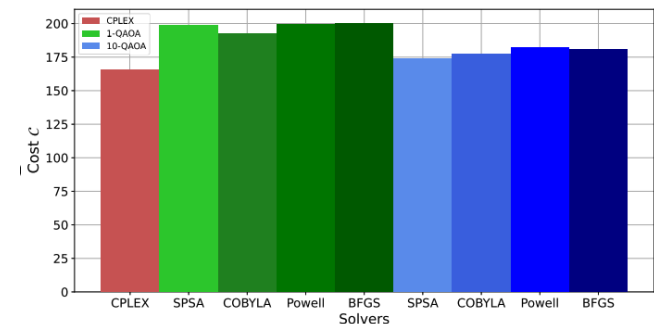
分析結果

- $p = 1$ のとき (図の緑色)
 - どの古典的optimizerを用いてもCPLEXの結果より悪い
- $p = 10$ のとき (図の青色)
 - QAOAの性能は向上し、CPLEXで得られたコストに近づいた
- Optimizerの効果
 - $p = 1$ のとき、4つの古典的optimizerの結果に大きな違いなし
 - $p = 10$ のとき、SPSAとCOBYLAが他の古典的optimizerよりも良い結果

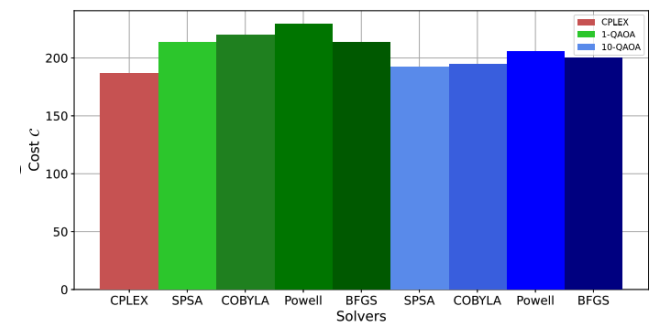
インスタンス (4, 2)



インスタンス (5, 2)



インスタンス (5, 3)



V . Discussion and Conclusion

Discussion

QAOAの性能とパラメータ p の影響

- 実験より実現可能な解を得られる確率が、量子回路の深さ p の増加とともに向上することが示された
- しかし、一般的に有限の p の値では元の組み合わせ最適化問題の最適解が保障されるわけではない
 - QAOAが断熱時間発展パスを p ステップで推測するため、選択された p では十分な推測を生まない
 - エネルギーランドスケープに新たな局所最適解が生まれ、古典的optimizerがそこに収束

QAOAに用いる古典的optimizerについて

- 変分パラメータを最適化する古典的optimizerの種類もQAOAの性能に影響
- 勾配ベースのoptimizerでは、変分パラメータの勾配の大部分が0になり訓練が停滞する可能性あり
 - $p = 10$ の実験でSPSAやCOBYLAのような勾配を使用しないoptimizerがより性能を示した理由の1つ？

Discussion

QAOAの課題：

p の値が増加すると、最適なパラメータを探索するためのトレーニング時間があらゆるoptimizerで指数関数的に長くなる。最適化するパラメータ数は線形に増加。

QAOAの性能向上のための改善方法

- 良い初期値の使用で最適化時間を大幅に短縮可能
 - 例1： $p - 1$ 層の最適化パラメータを p 層の初期値として使用
 - 例2：Parameter concentration → 小問題の解を大問題に転用
- 解空間を制限するために標準のミキサーハミルトニアンとは異なるミキサーを用いる
- 一方、NISQによるノイズの影響は大
→ノイズの多いQAOAの振る舞いを特徴づけることは、ノイズ耐性向上に不可欠

Conclusion

- QAOAはVRPのような組み合わせ最適化問題を解くために利用できる可能性がある
- 小さな問題インスタンスに対してはVRPを解くことができるが、より大きな問題インスタンスを解くのは困難
- 実用的なインスタンスのエンコードには数百万オーダーの量子ビットが必要
- 大きな p での古典的な変分パラメータ最適化が困難
→小規模問題でも、QAOAの速度向上は見られなかった

所感

- 卒論で最適化に触れなかったもので、本論文を通して研究室のメンバーがよく話しているVRPの気持ちが少しわかった気がする
- このくらいの問題インスタンスならステップ数 p はこの程度必要といった指標あると嬉しい
- ハード側の問題でハイブリッドな手法になってしまうことや実験でも古典的ソルバーに対しての十分な優位性が示せていなかったのが残念
→ 今後より高性能な量子コンピュータが実現すれば、量子最適化が主流に？
- GPTによる大まかな理解と熟読する理解の差がまだ感じられてよかった
- いくつか図の中に自分の理解と一致しない点が見られた
 - Fig. 4 (d)はインスタンス(5, 3)のVRPの解とあるが、図自体はFig. 4 (b)と同じもの？
 - Fig. 2 の回路図中の $e^{-i\hat{H}_{cost}\gamma}$ と $e^{-i\hat{H}_{mixer}\beta}$ のブロックとその詳細なゲート構造の対応が逆？

参考文献

- QunaSys. Quantum Native Dojo! – Qualacs. 2019. <https://dojo.qualacs.org/ja/latest/> (2025-05-12閲覧)
- MathWorks. 量子ゲートとは – MATLAB & Simulink – MathWorks. <https://jp.mathworks.com/discovery/quantum-gates.html> (2025-05-13閲覧)
- Qiita. 量子コンピュータ – ブロッホ球と各種ゲートの関係. 2021-01-07. <https://qiita.com/shnchr/items/15a42eaf971932230a50> (2025-05-14閲覧)