修士学位論文

データベース検索と勾配推定における量子アルゴリズムの最適化問題への応用

(Application to optimization problems by quantum algorithms for database search and gradient estimation)

東京大学大学院　理学系研究科
物理学専攻　浅井研究室

水原慎一

令和４年１月

概要

将来的に利用可能となるであろう量子コンピューターを用いた量子計算により、最適化問題における最適化過程において古典計算よりも優位な結果を得ることが本研究の目的である。現状の量子コンピュータで扱えるような量子計算の一つの形式として量子古典ハイブリッド型と呼ばれるものが主流であり、そこではパラメータの更新自体は古典コンピューターを用いて行われている。しかしそうしたパラメータの更新には指数的に計算量がスケールすると見込まれている。また一般の最適化問題としても本来最適化過程において得られるはずの量子計算の恩恵について知ることは非常に重要である。そのため本研究では最適化過程を量子コンピューター内で実行することを考え、そこで発生する優位性について議論する。

本研究では最適化過程としても頻繁に現れる探索による手法と関数の勾配の推定方法について取り上げる。より具体的には最適化過程への量子計算の応用として、最小値探索量子アルゴリズムと、量子計算を用いた効率的な勾配推定について考える。前者は、量子計算において最も重要なアルゴリズムの一つの Grover（データベース）探索の応用としても知られている。そこで前者を用いた探索によるパラメータの最適化として、パーセプトロンやフィードフォワードニューラルネットワークの古典的機械学習モデルを構築し、それらの学習を行う。シミュレーションによりその検証を行い、問題点の指摘を行う。一方後者として著名な Jordan のアルゴリズムを紹介する。その後ある非常に強い仮定のもとで整数係数・整数変数多項式関数に対し非常に少ない関数の評価回数で、位相の対称性を用いた新しい手法で勾配を推定する量子勾配推定アルゴリズムを提案する。量子勾配推定アルゴリズムを実験的に検証する試みがほとんどなされていない中、IBM の Quantum デバイスを用い部分的に検証した。同時に量子勾配推定アルゴリズムの考えられる応用例について議論する。

そしてそれまでの議論により得られた知見から、例えば勾配推定が可能な問題といった、実用上何らかの局所的構造を持った問題に対しサンプリングによる量子計算の恩恵はどの程度得られるのかといった問いが生まれる。それに対し定量的な議論を行なった研究は少ないと考えられる。そこで同問題に対し量子最小値探索を一般化しサンプリングにおける量子加速について、量子的 multi-start 法として定式化することで包括的に議論し幾つかの問題点と結論を得る。その上で勾配に関する量子アルゴリズムを組み合わせることで量子計算にさらなる優位性が発生することを、明示的に説明する。

最後にまとめると、量子勾配推定において考案した位相の対称性の利用が可能であることがわかった。また量子的 multi-start 法として定式化を行い実用上の問題を踏まえた上での包括的な議論を与えたが、サンプリング問題に対する一般的な問題としては本質的に大きな改善がなされるとは考えにくい。そのため本研究から得られた帰結として、量子計算における情報の符号化と特定の問題構造を適切に利用した量子計算が今後も重要な研究対象になると考えられる。

目次

図目次

表目次

第 1 章

序論

1.1 背景

量子計算 (Quantum Computing,QC)[1] とは、量子力学が対象とするような重ね合わせ状態や量子もつれといったミクロな物理現象を用いることで、従来の古典計算機では現実的な計算時間やリソースで解くことが困難とされる問題に対し有効であることが期待されている計算手法である。その最たる例が Shor の素因数分解アルゴリズム [2, 3] であり、現在知られている素因数分解のための最良の古典アルゴリズムに対し指数的な加速が量子計算により可能であることが示されている。

過去 10 年の量子計算にまつわる潮流としては、2011 年の D-Wave 社による世界初の商用型量子コンピュータの発表 [4] や 2019 年の Google 社による量子超越性[*1]の達成の主張 [5] を代表して、量子計算の理論的研究だけでなく実デバイスの開発競争が活発になっている。現在は IBM 社のクラウド経由の量子コンピュータ [6] を筆頭として、誰でも手軽に量子計算機にアクセスできる時代が到来しつつあると言える。

一方生物の神経回路を模した人工ニューラルネットワークに関する研究は、その誕生と誤差逆伝播法の発明を主な分岐点として隆盛と衰勢を繰り返してきた。原因は多層にすることで学習の勾配が消失する現象（勾配消失現象）と、後発の機械学習に比べ理論的な枠組みに欠けていた点である。そうした理由により、多層ニューラルネットに対する関心は薄れていった[*2]が、2006 年の Hinton らのディープビリーフネットワークの成功を皮切りにして [7]、以後深層学習（ディープラーニング）に関する研究が爆発的に進展した。近年の発達の理由は、理論的研究の進展のみならず、インターネットの発達と安価なデバイスの普及に伴い豊富な学習データを容易に収集しやすくなった [8] ことと、計算機の計算能力の飛躍的上昇によるところも大きい。

そうした中で量子計算による機械学習アルゴリズムの改善の可能性について研究者のみならず、企業や政府機関など幅広い組織からも注目を受けている。考えられる “量子” 機械学習 [9] の一つの恩恵は古典機械学習モデルの学習プロセスの改善とされており、それは最適化問題における、最適化過程への量子計算の応用の可能性へと帰着する。

しかし NISQ (Noisy-Intermediate Scale Quantum) 時代 [10] と呼ばれる現状においては、誤り耐性量子コンピュータ（fault-torelant quantum computer）[11] に必要とされる量子誤り訂正 [12] のために必要な数の量子ビットを用意することができない。そのため、ここ 5 年の量子機械学習としては、変分量子回路を用いた、量子・古典ハイブリッド型のアルゴリズム [13] が盛んとなっている。それは、主に量子並列性に起因する量子状態の指数的な表現能力の高さを利用しつつも、学習プロセス自体は観測に基づき古典計算機を用いた学習パラメータの更新を行う。しかし学習過程では古典コンピューターを用いた方式であるため、勾配消失問題 *Barren plateaus*[14][*3]が将来的に顕

[*1] ある問題に対し、どのような古典コンピュータを用いても現実的な実行時間で解けないが、量子デバイスを用いることでそれが可能となることを実証すること。

[*2] 例外的に畳み込みニューラルネットワークは 1980 年代には成功していたが、多層ニューラルネットの研究全般の衰退につられる形で下火になっていった。

[*3] 勾配が量子ビット数に関し指数的に減衰し指数回の観測が必要とされる問題。利点であるはずの量子状態の指数的な表現能力に起因する。

著となると考えられている。そうした背景において、本来量子計算が最適化過程にもたらすはずの恩恵について知ることは将来的に非常に重要であると考えられる。

本研究では、最適化問題における最適化過程の量子計算の応用可能性または量子優位性[*4]について、実機でのスケーラビィティも調査しつつ主に探索問題、勾配推定、局所最適化の観点から述べる。

1.2　本研究の位置付け

第 1.1 節において、本研究の主題を述べた。ここではより具体的に、概要で説明した本研究の成果と先行研究との関連性について説明する。

古典機械学習モデルを量子計算によって再現する際、最も初めに考えられるべきは古典計算と同じく量子状態におけるビットの列を数とみなすことで数値計算を行う方法である。このようなビットの計算基底への情報の符号化自体は古典的であり、量子機械学習としてもアイデア自体は 20 年ほど前から存在する [15] が、実際に学習モデルの構成に必要な量子操作を一から構成した例は少なく、せいぜい最も単純なモデルであるパーセプトロン程度しかその対象とされていない [16]。そのため本研究では実際に、パーセプトロンと隠れ層 1 つの順伝搬型ニューラルネットワークのモデルを上述の符号化により構成した。その上で最小値探索量子アルゴリズムの使用を前提に、量子振幅増幅アルゴリズムを用いて学習を行い問題点について指摘する。結果的に特定の問題に特化した量子計算の構築が重要であると結論づけ、本研究最後の到達点である局所最適化が可能な問題への量子振幅増幅の応用の足掛かりとする。

次に最適化問題において目的関数の勾配情報が利用できる場合、一般にそうでない場合に比べ最適化が容易であることが知られている。関数の勾配を用いた最適化手法を勾配法といい、勾配法により一般に関数最小化が可能で例えば Fitting や機械学習における学習すなわちコスト関数の最小化に応用可能であるが、扱うデータ量の増大に伴い関数の次元に関する空間の指数的増大、局所的最小点や停留点の増大、勾配自体の計算量の増大が起き、先述した勾配消失問題も含めて勾配計算に関連した研究が盛んに行われている [17, 18, 19, 20, 21]。本研究では勾配の推定に関し効率的な量子計算 (量子勾配推定と本研究では呼ぶこととする) を主眼とし、代表的な Jordan の量子勾配推定アルゴリズムを主としつつも、位相の対称性を用いる新たな手法を用い勾配推定の量子アルゴリズムを考案した。それにより整数変数・整数係数多項式関数に対し関数がブラックボックスであると仮定して、$O(1)$ 回のクエリにより勾配のビット表現をあるビット数まで得ることで、勾配の絶対値が十分小さい場合適切に勾配を得ることができる。そして実機による量子勾配推定が現状なされていない中、実際に超伝導量子コンピュータである IBM Quantum デバイスを用いて同アルゴリズムを部分的に検証することに成功した。そして考案したアルゴリズムを含め、量子勾配推定の応用可能性自体についても議論を与える。

このように現実的には最適化問題自体に勾配が推定できるような、何らかの構造が存在していたり仮定を用いることができる状況が多いことから、量子計算を特定の問題に特化する方向性が適切である。実際最小値探索量子アルゴリズムは、非常に多くの量子アルゴリズムに適用可能であって広範に利用されていることが多い。しかし一方で、特定の最適化問題の形式に依拠した議論が非常に多く、一般の問題への適用を前提に定量的に議論された例は少ない。特に一つの一般的問題として、局所最適化が可能な問題が存在するため、最小値探索量子アルゴリズムの同問題への応用可能性を定量的に議論する。適用の結果としては、局所解を避けるためにサンプリングを複数回とる古典的 multi-start 法（**Algorithm 9** 参照）に対する、サンプリング部分の加速としてまとめる。それを最終目標として先行研究 [22, 23] をもとにし、本研究では具体例を踏まえつつ可能な限り一般化した形式において、確率的局所最適化と量子最小値探索を組み合わせることで得られる量子加速について包括的に議論する。

その枠組みの中で具体例として勾配推定量子アルゴリズムと組み合わせることの可能性について議論する。量子勾配推定と量子最小値探索を組み合わせるアイデアは、2005 年に文献 [22] で考案されたが、これを現実問題を踏まえ

[*4] 特定の計算タスクにおいて量子デバイスが最良の古典計算機を大きく上回る性能を発揮することを示すこと。

た上で量子的 multi-start 法という広い枠組みで理解することを可能とする。このように特定の構造に対しても、量子的 multi-start 法という一般化した形式を前提としたより詳細な議論を行うことを可能にする。

1.3　量子 AI という俯瞰視点での本研究

第 1.1 節で機械学習と量子計算の融合分野として量子機械学習の紹介をし、探索問題、勾配推定の視点から 最適化過程における量子計算による恩恵を本研究の主題として説明した。第 1.2 節で先行研究との関連性の中で本研究成果を説明した。その他にも本研究を特徴づけるにあたって、機械学習と量子計算を組み合わせた新興分野としての前述の量子機械学習に注目が集まっている中、連鎖的に関心が高まっている “量子 AI” が挙げられる。よってここでは量子機械学習よりもさらに広い視点である量子 AI という視点で、本研究を説明する。

それにあたって量子 AI または量子人工知能 (Quantum Artificial Intelligence, QAI) について説明する必要がある。本研究では量子 AI に関する文献 [24] での議論を引用またはそれに基づき説明を行う。

量子 AI を語るにあたって量子計算と AI についての説明が必須である。量子計算については、第 1.1 節で量子現象を利用した、通常の計算よりも有効であると期待されている計算方法と説明した。具体的な例は、後の本文を参照されたい。一方 AI についてはここで説明が必須である。AI の分野は、主にコンピューターには困難であるが人間にとっては容易な問題を解くことに焦点を当てた、様々な手法・技術の集合体となっている。AI とは、人間を含む動物が持つとされる知能を機械によって発現させる技術、人間の知的活動を計算機に代替させる技術などとされる。その実現のための最重要クラスが学習とされ、そうした学習問題に対してはアルゴリズム的観点で見ればパターン認識の研究から端を発した、機械学習の分野として扱われて大きな成功を収めた。現代的な機械学習の分野は、教師つき（データ分類）・教師なし（データクラスタリング）・強化学習（相互作用からなる学習）の 3 つに大きく分類されるが、それら機械学習を内包する AI という分野に対する現代的な見方は、エージェントと環境という概念に基づいている。エージェントとは、環境の存在を前提に環境と結び付けられた存在であり何らかの行動を行うものとされ、学習はエージェントと環境との相互作用として理解される。そのような定式化の上で AI の究極的目標は、知的エージェントが機械学習またはその他の AI の技術で人間が扱うような知的タスクを解決する、汎用 AI(Artificial general intelligence, AGI) の実現とされるが、その実現にあたって何が必要かでさえ共通認識が形成されていない。さらにはそもそも何をもって知的であるとするかという問いは多くの科学者と哲学者を悩ませてきた。

このように AI に関し、知能とは何か、真の AI はどのように達成されるかといったシンプルな問いに答える事が困難を極めるように、“量子 AI とは何か” ということを考えることは明らかに得策ではない。しかしながら、量子 AI の概念は少なくとも二つの要素を持つと考えられている。一つは、AI 分野の目標と同じくまさに知能という概念を体現したものであるはずである。二つ目は、より実践的な立場に基づき、知能という概念自体の一般化は要さずに、量子効果をより知的なエージェントの生成に役立つ可能性としてみなす解釈 (quantum-enhanced AI) である。そうした理解の上で、量子計算と AI の分野横断的な相互作用の可能性として（具体的な説明は同文献 [24] を参照してもらうこととして)、量子 AI に関し以下の 4 つの方向性が考えられている。

量子現象への機械学習の応用（Applications of ML in quantum physics）

- 推定・測定（Estimation and metrology）
- 量子系の制御（Quantum control and gate design）
- 量子実験の制御と AI を利用した物理現象の理解（Controlling quantum experiments, and machine-assisted reaserch）
- 物性物理学や多体系の物理学（Condensed matter and many body physics）

量子計算による機械学習の改善（Quantum enhancements for ML）

- 量子的パーセプトロン・ニューラルネットワーク（Quantum perceptrons and neural networks）
- 量子計算に基づく学習理論（Quantum computational learning theory）
- 量子計算による学習容量の向上（Quantum enhancement of learning capacity）
- 学習における量子加速（Quantum computational algorithmic speedups for learning）

機械学習タスクの量子的な拡張（Quantum generalizations of ML-type tasks）

- 量子データの機械学習（Quantum generalizations: machine learning of quantum data）
- 量子過程の（量子的）学習（(Quantum) learning of quantum processes）

量子的な学習エージェントと量子 AI の構成要素（Quantum learning agents and elements of quantum AI）

- 量子計算により拡張された、相互作用を通じた学習（Quantum-enhanced learning through interaction）
- 量子力学に基づいたエージェント・環境モデル（Quantum agent-environment paradigm）
- 量子 AI に向けた議論（Towards quantum AI）

可能な量子 AI への取り組みとして、本研究はまさに 2 番目の方向性に該当すると考えられる。より細かく言えば、学習を最適化として捉えた、探索と勾配推定に関した最適化（過程）における量子加速に関する研究であると位置付けられる。このように本研究は、量子 AI の実現に向けた、実践的な取り組みの一つとして解釈することも可能であり、本研究がより一層重要であると言える。

1.4　本論文の構成

第 1 章において量子計算にまつわる背景と本論文の構成について触れる。第 2 章において量子計算の導入と最適化にまつわる代表的なアルゴリズムとして、量子振幅増幅（Grover 探索）アルゴリズム、最小値探索アルゴリズム、Jordan の勾配推定量子アルゴリズムについて紹介する。第 3 章では量子回路内において基本的な古典的機械学習モデルであるパーセプトロンと隠れ層 1 つの順伝搬型ニューラルネットワークによる 2 値分類タスクのための学習（最適化）を行う。第 4 章では新たに、整数係数・整数多項式関数に対する勾配推定量子アルゴリズムを考案・提案し、実機を用いた結果について説明したのち同アルゴリズムと量子勾配推定アルゴリズム全般の応用先について触れる。第 5 章では、量子最小値探索アルゴリズムを一般化しさらに確率的局所最適化アルゴリズムと組み合わせた際の優位性について具体例を交えつつ議論する。その上で局所最適化として量子勾配推定を行ったときの優位性について議論する。第 6 章では本研究により得られた結果についてまとめる。

付録 A では計算量の記法、付録 B では量子回路内における数の表現方法と量子フーリエ変換に基づいた初等演算の構成方法、付録 C では本論文で対象とする最適化問題についてまとめている。必要に応じて参照されたい。

第 2 章

量子計算

以下基礎的な量子力学の知識を前提とする。

量子状態の初期化やユニタリ変換、測定といった量子系に対する操作は量子操作 (quantum operation) という。そして量子計算を用いたアルゴリズム（量子アルゴリズム）を記述するにあたって量子回路を用いる方法が存在する。量子回路モデルにおいては以下で説明する量子回路の構成要素を用いて、量子ビットの初期化、量子ゲートと測定の反復またはその他の量子操作として量子アルゴリズムを記述する。

2.1 量子ビット

初めに古典ビットについて振り返る。ビットとは情報の単位であり、１ビットによって２次元の古典系を表現することができる。例えば２次元古典系を

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix},\ |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{2.1.1}$$

と表すことができる。ビットの物理的表示としては２つのフリップフロップな状態の表示に対応し、例えば電子回路内の異なる２つの電位値であったり、異なる２つの光強度がそれにあたる。

一方量子計算においては量子効果である状態の重ね合わせを利用することができる。量子ビット (qubit, quantum bit) とは、２次元の量子系を記述する情報の単位であり、量子ビットの一般的状態は

$$c_0 |0\rangle + c_1 |1\rangle = \begin{pmatrix}c_0\\c_1\end{pmatrix} \tag{2.1.2}$$

と規格化条件 $|c_0|^2 + |c_1|^2 = 1$ を満たす複素数 c_0, c_1 を用い $|0\rangle, |1\rangle$ の重ね合わせ状態として表現することができる。物理的には光子の偏極状態やスピン $\frac{1}{2}$ の粒子のスピンなどとして表現できる。

古典的な情報の表現として例えば 000110 といったビットの列を用いるが、量子計算においては量子ビットのテンソル積

$$|0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \tag{2.1.3}$$

のように表現することができる。これは $2^6 = 64$ 個の成分を持つ基底によって

$$|000110\rangle = \begin{pmatrix}0 \leftarrow |000000\rangle\\0 \leftarrow |000001\rangle\\\vdots\\0 \leftarrow |000101\rangle\\1 \leftarrow |000110\rangle\\\vdots\\0 \leftarrow |111110\rangle\\0 \leftarrow |111111\rangle\end{pmatrix} \tag{2.1.4}$$

とベクトルとして表すことができる。こうしたビット列を古典計算と同じく二進数表示と対応させることで、$|000110\rangle = |6\rangle$ としばしば簡易的な表現を用いる。より一般に n 個の量子ビットを持つ量子系は 2^n の次元を持ち、それに属する一般的な量子状態 $|\psi\rangle$ は、計算基底と呼ばれる正規直交基底 $\{|i\rangle\}_{i=0}^{2^n-1}$ を用いて

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle \tag{2.1.5}$$

とベクトルとして表すことができる。ここで $c_i \in \mathbb{C}$ $(i = 0, \ldots, 2^n - 1)$, $\sum_{i=0}^{2^n-1} |c_i|^2 = 1$ とする。この状態の並列性から量子計算は古典計算に比べビット数に関し指数的な表現能力があるとされる。

量子回路モデルにおいて 1 量子ビットは 1 本の水平な配線として描かれる。n 量子ビットを用いる量子回路なら図 2.1 のように n 本の配線を縦に並べる。量子ビットが $|x_1\rangle |x_2\rangle \ldots |x_n\rangle$ と初期化されている場合、配線を上から順に量子ビットを対応させ各配線の左端にその状態を記す。そして配線に量子操作を加える（次節で説明する）ことで具体的な量子計算を記述する。量子操作後の状態は量子回路図において配線の右端に記すことで表現する。

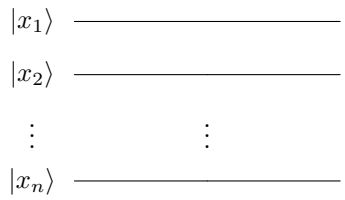


図 2.1: 量子回路図における量子ビットの初期化の表現

単一または複数の量子ビットからなる量子系を量子レジスター (quantum register) と呼ぶ。量子回路図において特定の量子レジスターを指し示す際、配線の左端にその呼び名を記す。配線の左端に状態の初期化が明記されておらず特に指定がない場合、基底状態 $|0\rangle$ に初期化されているとされる。

複数の配線をまとめて表現する際は図 2.2 のように描く。必要に応じてその本数を図 2.2 内の斜線付近に書き入れる。

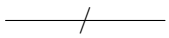


図 2.2: 量子回路図における複数の配線の表現

2.2　量子ゲート

量子系に対するユニタリ変換は、すなわち量子ビットまたは量子レジスタに対するユニタリ変換であり、量子回路モデルにおいてこのユニタリ変換を量子（論理）ゲート (quantum logic gate) と呼ぶ。量子レジスタ qr に変換 U を行う量子操作の量子回路図上の表現として、図 2.3 のようにターゲットとなる複数の量子ビット（配線）に長方形を描きその内部に演算名 U を書き入れる。

量子レジスター qr_1 にユニタリ変換 U_1 を行った後に量子レジスター qr_2 にユニタリ変換 U_2 を行うといった、複数の量子操作に時間的順序が存在する場合は、量子回路図において左から右の向きに時間軸が存在していると考えて左から順に量子ゲート U_1, U_2 を描く。

実際に量子アルゴリズムを構成するにあたって所望のユニタリ変換つまり量子ゲートがどのようにして構成できるかが重要となる。関連する重要な事実として、任意のユニタリ演算を任意の精度で近似するに十分なゲートの集合が

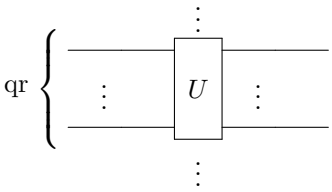


図 2.3: 量子回路図における量子レジスタ qr に対するユニタリ変換 U の表現

存在する。そうしたゲート集合は量子計算において普遍的 (universal) であるという。普遍的量子計算についてここでは深く触れないが、以下で説明する代表的な量子ゲートの一部を用いればそれが可能である。

量子ゲートはユニタリ変換であるからそれを表現するには行列を用いる。行列表現の際の基底としては、計算基底を選択し量子状態のベクトル表現の式 (2.1.4) に則って、n 量子ビットに対する量子ゲート U の行列表現は $N = 2^n$ として

$$U \equiv \begin{pmatrix} U_{00} & \cdots & U_{0N-1} \\ \vdots & \ddots & \vdots \\ U_{N-10} & \cdots & U_{N-1N-1} \end{pmatrix} \equiv \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} U_{ij} \, |i\rangle\langle j| \tag{2.2.1}$$

とし、行列要素 U_{ij} $(0 \leq i, j \leq N-1)$ は計算基底 $|i\rangle\langle j|$ に対応する。テンソル積の定義は通常の量子力学に従い省略する。

2.2.1 1量子ビットゲート

単一量子ビットに対するゲートを 1 量子ビットゲートといい、重要なものを紹介する。

単位 (Identity) ゲート σ_0

$$\sigma_0 \equiv I \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \equiv \text{—}\boxed{I}\text{—} \tag{2.2.2}$$

パウリ (Pauli) ゲート σ_i $(i = 1,\ 2,\ 3)$

$$\begin{aligned} \sigma_1 \equiv X \equiv \text{NOT} \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \equiv \text{—}\boxed{X}\text{—} \equiv \text{—}\oplus\text{—}, \\ \sigma_2 \equiv Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \equiv \text{—}\boxed{Y}\text{—}, \ \sigma_3 \equiv Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \equiv \text{—}\boxed{Z}\text{—} \end{aligned} \tag{2.2.3}$$

ここで交換関係、反交換関係として関係式

$$[\sigma_i,\ \sigma_j] = 2i\epsilon_{ijk}\sigma_k \tag{2.2.4}$$

$$\{\sigma_i,\ \sigma_j\} = 2i\delta_{ij}I \tag{2.2.5}$$

が成立する。ここでクロネッカーのデルタ δ_{ij}、レヴィ＝チヴィタの記号 ϵ_{ijk} とアインシュタインの縮約規則を用いた。

アダマール（Hadamard）ゲート、H ゲートは

$$H \equiv \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \equiv \text{—}\boxed{H}\text{—} \tag{2.2.6}$$

である。自然数 n として n 個のアダマールゲートを用いた変換 $H^{\otimes n}$ をアダマール変換（Hadamard transform）といい、一様な振幅の重ね合わせ状態を用意するのに非常によく用いられる変換である。

回転ゲート $R_x,\ R_y,\ R_z$

$$R_x(\theta) \equiv \exp\left(-i\frac{\theta}{2}X\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \equiv \text{—}\boxed{R_x(\theta)}\text{—} \tag{2.2.7}$$

$$R_y(\theta) \equiv \exp\left(-i\frac{\theta}{2}Y\right) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \equiv \text{—}\boxed{R_y(\theta)}\text{—} \tag{2.2.8}$$

$$R_z(\theta) \equiv \exp\left(-i\frac{\theta}{2}Z\right) = \begin{pmatrix} \exp\left(-i\frac{\theta}{2}\right) & 0 \\ 0 & \exp\left(i\frac{\theta}{2}\right) \end{pmatrix} \equiv \text{—}\boxed{R_z(\theta)}\text{—} \tag{2.2.9}$$

位相ゲート P は

$$P(\theta) \equiv e^{i\frac{\theta}{2}}R_z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & \exp\left(i\theta\right) \end{pmatrix} \equiv \text{—}\boxed{P(\theta)}\text{—} \tag{2.2.10}$$

であり、行列表現を見れば位相ゲートと回転ゲート R_z はグローバル位相を除いて一致している。

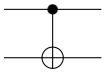
SX ゲート、$\sqrt{X}$ ゲート

$$\mathrm{SX} \equiv \sqrt{X} \equiv e^{i\frac{\pi}{4}}R_x\left(\frac{\pi}{2}\right) = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix} \equiv \text{—}\boxed{\mathrm{SX}}\text{—} \equiv \text{—}\boxed{\sqrt{X}}\text{—} \tag{2.2.11}$$

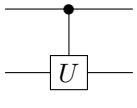
2.2.2　2 量子ビットゲート

2 量子ビットに作用する量子ゲートを 2 量子ビットゲートと呼び、ここでも代表的なものを紹介する。

制御 X ゲート (Controlled-X gate)、CX ゲート、CNOT ゲートは制御（control）ビット、標的（target）ビットを持ち、標的ビットを条件付きでビット反転させる。

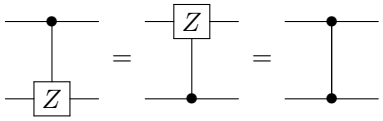
$$\mathrm{CX} \equiv |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv$$  (2.2.12)

制御 U ゲート（Controlled-U gate）は CX ゲートの標的ビットに施すゲート操作 X を任意の 1 量子ビットゲート U へと拡張したものであり、i 番目に位置する制御ビットと j 番目に位置する標的ビットに作用するとして

$$C_j^i[U] \equiv |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix} \equiv$$  (2.2.13)

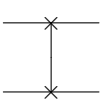
のように記述される。（より一般に標的ビットが複数の量子ビットであれば U に結合する配線を増やして表現する。）

制御 Z ゲート（Controlled-Z gate）、CZ ゲートは、制御ビットと標的ビットを入れ替えても等価なゲートであるため特殊な記法

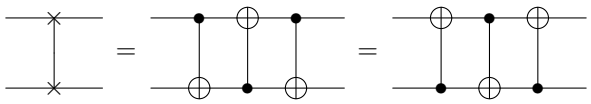
$$\mathrm{CZ} \equiv |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \equiv$$  (2.2.14)

を用いる。

SWAP（スワップ）ゲート、交換ゲートとは

$$\mathrm{SWAP} \equiv |0\rangle\langle 0| \otimes |0\rangle\langle 0| + |0\rangle\langle 1| \otimes |1\rangle\langle 0| + |1\rangle\langle 0| \otimes |0\rangle\langle 1| + |1\rangle\langle 1| \otimes |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \equiv$$  (2.2.15)

であり、３個の CX ゲートで構成することが可能である。

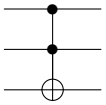


(2.2.16)

2.2.3　その他のゲート

トフォリ (Toffoli)、CCX ゲート

$$\mathrm{CCX} \equiv |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes \mathrm{CX} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \equiv$$



(2.2.17)

2.2.4　測定

量子系に対する測定操作は、量子回路モデルでは配線の右端に特殊な記号を描き、図 2.4 のように表現することができる。この際測定における基底は計算基底とされる。

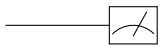


図 2.4: 量子回路図における量子ビットに対する測定の表現

また本論文中では現れないが、測定結果など古典情報に基づき量子操作を行う場合があり、そうした古典情報の通信は２重の配線を用いて送信元と受信先を直線的に結ぶ。

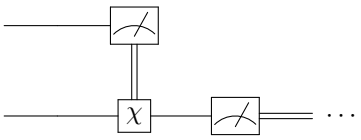


図 2.5: 量子回路図における古典情報の伝達の表現例

2.3　量子回路の例

ここまで量子回路モデルによる量子計算の記述法を説明してきた。そうした個々の量子回路を特徴づける量として幾つかの尺度が存在する。一つは量子回路を基本的なゲート集合で記述した時のゲート数（サイズという)、二つ目は量子回路の深さ（時間的に同時に実行できる複数のゲートは鉛直方向に見て同じ１つの層に属すると見なし、水平方向で数えた時の層数)、量子ビット数などがある。ゲート数により量子回路の複雑性を評価する際は、基本的に実装コストが比較的かかるとされる CX, CCX ゲートが比較的用いられる。

その上でここでは本論文に関連する重要な量子回路の例として量子フーリエ変換のための回路と Draper の加算回路を紹介する。

2.3.1 量子フーリエ変換

量子フーリエ変換 (Quantum Fourier Transform, QFT)[25] は、量子計算において最も強力で重要な変換の一つである。量子フーリエ変換は、自然数 $n, N = 2^n$ として n-qubit のヒルベルト空間 $\mathcal{H}_N$ 上での、以下のユニタリ変換で定義される。

$$\mathrm{QFT}\colon\ |j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(2\pi i \frac{jk}{N}\right) |k\rangle \tag{2.3.1}$$

ここで $\mathcal{H}_N$ の N 個の正規直交基底を $\{|k\rangle\}_{k=0}^{N-1}$ とし、定義内における $|j\rangle$ はそれらのうちのいずれか一つである。量子フーリエ変換はユニタリー変換であるから逆変換が存在し、それは逆量子フーリエ変換 (Inverse Quantum Fourier Transform, IQFT) として以下で定義される。

$$\mathrm{IQFT}\colon\ |j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(-2\pi i \frac{jk}{N}\right) |k\rangle \tag{2.3.2}$$

一般の量子状態 $|x\rangle \in \mathcal{H}_N$ に対する量子フーリエ変換の作用は、同じ正規直交基底 $\{|k\rangle\}_{k=0}^{N-1}$ を用いて

$$\mathrm{QFT}\colon\ |x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle \mapsto |y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle \tag{2.3.3}$$

と表される。ただし

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left(2\pi i \frac{jk}{N}\right) \tag{2.3.4}$$

である。

量子フーリエ変換は、古典計算における離散フーリエ変換の逆変換をヒルベルト空間内の量子状態に施したものとして捉えることができる。それは以下のようにして確認できる。まず離散フーリエ変換 (Discrete Fourier Transform, DFT) とは、自然数 N に対し

$$\mathrm{DFT}\colon \mathbb{C}^N \ni \boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1}) \mapsto \mathbb{C}^N \ni \boldsymbol{z} = (z_0, z_1, \ldots, z_{N-1}) \tag{2.3.5}$$

でありかつ各 z_k が

$$z_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left(-2\pi i \frac{jk}{N}\right) \tag{2.3.6}$$

で定義されるものである。$N = 2^n$ における離散フーリエ変換の逆変換 (Inverse Discrete Transform, IDFT) は、量子フーリエ変換における y_k を用いて

$$\mathrm{IDFT}\colon \mathbb{C}^N \ni \boldsymbol{x} = (x_0, x_1, \ldots, x_{N-1}) \mapsto \mathbb{C}^N \ni \boldsymbol{y} = (y_0, y_1, \ldots, y_{N-1}) \tag{2.3.7}$$

で定義され確かに対応が見て取れる。以下 $N = 2^n$ に限定して話を進める。

次に量子フーリエ変換を単純なゲートによって構成するために、計算基底状態 $|j\rangle$ に対する量子フーリエ変換の積表現

$$\mathrm{QFT}\,|j_1 j_2 \ldots j_n\rangle = \frac{1}{\sqrt{N}} \left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n} |1\rangle\right) \tag{2.3.8}$$

の成立を確認する。ここで自然数 j に対する 2 進数表現 $j \equiv j_1 j_2 \dots j_n \equiv \sum_{k=1}^{n} j_k 2^{n-k}$ と 2 進小数としての表示 $0.j_k j_{k+1} \dots j_l \equiv \sum_{m=k}^{l} j_m 2^{k-1-m}$ を用いた。これが量子フーリエ変換の定義と完全に等価であることは、定義式から逐次的に示せる。実際、

$$\begin{aligned}
\mathrm{QFT}\,|j\rangle &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(2\pi i j \sum_{l=0}^{n-1} k_l 2^{-l}\right) |k\rangle = \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \exp\left(2\pi i j \sum_{l=0}^{n-1} k_l 2^{-l}\right) |k_1 \cdots k_n\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} \exp\left(2\pi i j k_l 2^{-l}\right) |k_l\rangle = \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(\sum_{k_l=0}^{1} \exp\left(2\pi i j k_l 2^{-l}\right) |k_l\rangle \right) \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(|0\rangle + \exp\left(2\pi i j 2^{-l} |1\rangle\right) \right) \\
&= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle \right) \otimes \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle \right)
\end{aligned}$$

であることから確かめられる。この積表現の各 qubit に着目すれば、位相部分に $j_1, \dots, j_n$ のいずれかが左端から見て一つずつ追加されている。これはすなわち、各 qubit ごとに対象となる qubit 以下の qubits との制御演算によって、段階的に $|j_1 \dots j_n\rangle$ からこの積表現へと変化させることができる可能性を示唆している。そしてそれは容易に実現可能である。

そのための以下の量子操作

$$V_1 \equiv U_n U_{n-1} \cdots U_1,\ U_i = \begin{cases} \mathrm{CP}_{n,i}\left(2^{i-n}\pi\right) \cdots \mathrm{CP}_{i+2,i}\left(2^{-2}\pi\right) \mathrm{CP}_{i+1,i}\left(2^{-1}\pi\right) H_i & (i \neq n) \\ H_n & (i = n) \end{cases} \tag{2.3.9}$$

とそれに連なる高々 $n/2$ 回の SWAP 操作

$$V_2 \equiv \mathrm{SWAP}_{1,n}\, \mathrm{SWAP}_{2,n-1} \cdots \mathrm{SWAP}_{\lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{2} \rfloor + 1} \tag{2.3.10}$$

によって、$\mathrm{QFT} = V_2 V_1$ として実現可能である。ここで $\mathrm{CP}_{i,j}$ は上から i 番目の qubit を制御 qubit、j 番目の qubit を標的 qubit とした制御位相ゲート、H_i は i 番目の qubit への Hadamard ゲート、$\mathrm{SWAP}_{i,j}$ は i, j 番目の qubits に作用する SWAP ゲートとした。V_1 による量子操作とそれによる量子状態の変化は図 2.6 に示した。

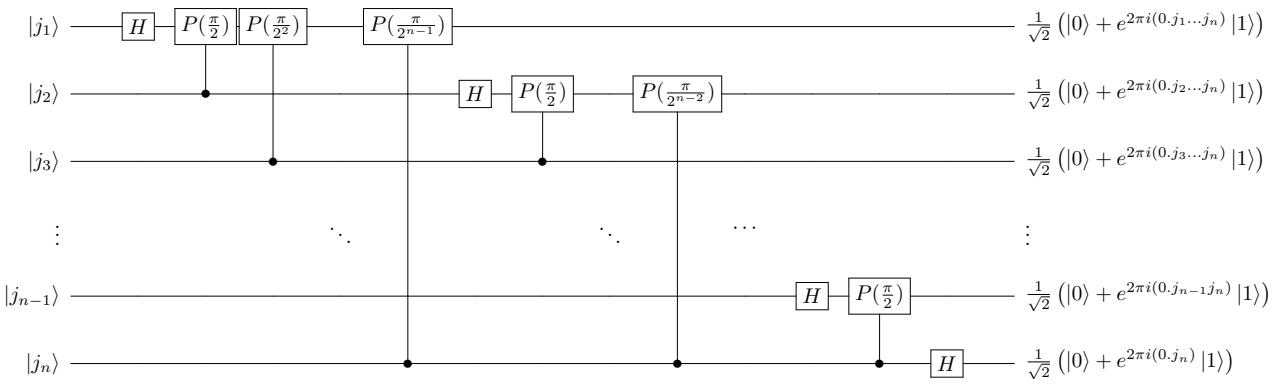


図 2.6: 量子フーリエ変換のためのゲート操作の一部 V_1 とそれによる状態変化

この具体的構成方法から量子フーリエ変換に必要なサイズを評価することができる。SWAP ゲートが 3 個の CNOT ゲートで構成できることから、必要な制御ゲートと Hadamard ゲート、CNOT ゲートの数の和は

$n+(n-1)+\cdots+1+3\lfloor\frac{n}{2}\rfloor = n(n+1)/2+3\lfloor\frac{n}{2}\rfloor = O(n^2)$ 個であり、$O(n^2)$ のサイズで量子フーリエ変換が構成できることがわかった。さらに量子回路内でデコヒーレンスが起きる場合においては、近似量子フーリエ変換 (Approximate Quantum Fourier Transform, AQFT) によって $O(n\log(n))$ のサイズで近似的に量子フーリエ変換が構成できることが知られている [26]。一方量子フーリエ変換の逆変換に対応する古典的離散フーリエ変換のための、最良の古典的アルゴリズムは高速フーリエ変換 (Fast Fourier Transform, FFT) であって $\Theta(n2^n)$ のサイズを持つ [27, 28]。そのため量子フーリエ変換を実問題に用いることで計算量を指数的に抑えることが可能であることが期待される。しかし注意すべきは、変換後の振幅の値は実際には観測するまで獲得することができず、仮に行うにも多数回観測が必要である点である。さらに量子フーリエ変換を適用できる量子状態を効率的に準備する手法も発見法的にしか理解されていない。しかし Shor の素因数分解 [2, 3] に代表されるように、それら欠点を補って余りある恩恵をもたらしうるので、量子フーリエ変換は量子計算において最も重要な変換の一つと言える。

2.3.2 Draper の加算回路

量子回路内で四則演算が実行できることが、複雑な量子計算をするための前提であると言える。ここでは四則演算のうち最も基本となる足し算を実行する回路である加算回路を考える。古典計算においては半加算器、全加算器を定義してそれらを組み合わせることで加算回路を構成する。しかしそれらは非可逆な回路となっているため、量子回路では実行できない。一方可逆な古典計算であれば量子回路で実行可能であるから、古典的手法を可逆な形式に変更し加算回路を構成する手法が存在する。しかしここでは、Draper の加算回路 [29] と呼ばれる、量子フーリエ変換を用いた加算回路の構成方法を述べる。

加算を行う 2 つの数として、n ビットの 2 進数表現された非負の整数 $a,\ b$ を考える。$|a\rangle|b\rangle \to |a+b \bmod 2^n\rangle|b\rangle$ とする加算回路に対応する量子操作の構成を目標とする。まず $|a\rangle$ のレジスタについて量子フーリエ変換を行い、

$$\mathrm{QFT}|a\rangle|b\rangle \equiv |\phi(a)\rangle|b\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}\exp\left(2\pi i a\sum_{l=0}^{n-1}k_l 2^{-l}\right)|k\rangle|b\rangle \tag{2.3.11}$$

を作る。次に $\mathrm{CR}_{j,k}$ を標的ビットが $|\phi_j(a)\rangle$ で制御ビットが $|b_k\rangle$ である制御回転ゲートとして量子操作

$$\bigotimes_{j=1}^{n}\bigotimes_{k=j}^{n}\mathrm{CR}_{j,k}\left(\frac{\pi}{2^{k-j}}\right)|\phi(a)\rangle|b\rangle \tag{2.3.12}$$

を行うと

$$|\phi(a+b)\rangle|b\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}\exp\left(2\pi i(a+b)\sum_{l=0}^{n-1}k_l 2^{-l}\right)|k\rangle|b\rangle \tag{2.3.13}$$

となる（図 2.7 参照）。さらに逆量子フーリエ変換を行うことで

$$\mathrm{IQFT}|\phi(a+b)\rangle|b\rangle = |a+b \bmod 2^n\rangle|b\rangle \tag{2.3.14}$$

が得られる。まとめると、この一連の量子操作を以下 U_{add} と名付ければ、

$$U_{\mathrm{add}}|a\rangle|b\rangle = |a+b \bmod 2^n\rangle|b\rangle \tag{2.3.15}$$

$$U_{\mathrm{add}} \equiv \left(\mathrm{IQFT}\otimes I^{\otimes n}\right)\left(\bigotimes_{j=1}^{n}\bigotimes_{k=j}^{n}\mathrm{CR}_{j,k}\left(\frac{\pi}{2^{k-j}}\right)\right)\left(\mathrm{QFT}\otimes I^{\otimes n}\right) \tag{2.3.16}$$

となる。

減算回路も全く同様に構成することが可能で、対応する量子操作を U_{sub} として以下のように定義する。

$$U_{\mathrm{sub}}|a\rangle|b\rangle = |a-b \bmod 2^n\rangle|b\rangle \tag{2.3.17}$$

$$U_{\mathrm{sub}} \equiv \left(\mathrm{IQFT}\otimes I^{\otimes n}\right)\left(\bigotimes_{j=1}^{n}\bigotimes_{k=j}^{n}\mathrm{CR}_{j,k}\left(-\frac{\pi}{2^{k-j}}\right)\right)\left(\mathrm{QFT}\otimes I^{\otimes n}\right) \tag{2.3.18}$$

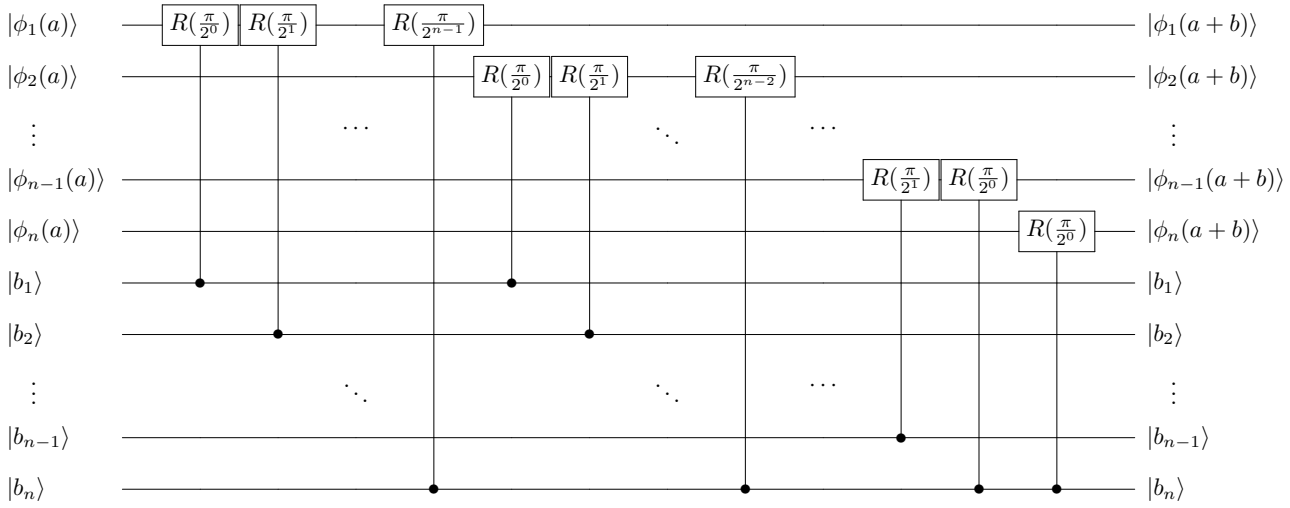


図 2.7: Draper の加算回路構成のための量子操作の一部

Draper の加算回路の特徴として量子フーリエ変換と異なり、式 (2.3.12) の部分を一部並列に実行することが可能で深さ $O(\log_2 n)$ にすることができる。また Draper の加算回路のサイズは量子フーリエ変換とその逆変換そして $\frac{n(n+1)}{2}$ 回の制御回転ゲートを用いるため、$O(n^2)$ となっている。一方古典的加算回路のサイズは $O(n)$ であることがわかっているので、それよりも劣る。それでもなお古典計算では不可能な手法として重要視されている。例として古典加算回路において、もし構成ゲートの一部でも欠損や問題が発生すれば全体として加算回路とは全く異なる挙動をするが、そうした場合でも Draper の加算回路では近似的に正しい動作をするという特徴がある。

2.4 最適化に関する量子アルゴリズムの例

アルゴリズムにおける各ステップで量子計算を用いる量子アルゴリズム (quantum algorithm) が量子計算の一つの重要な柱となる。

そうした何らかの量子アルゴリズムが有効または効率的であるか調べるには、アルゴリズムの計算量を評価する必要があり、そのための方法が幾つか存在する。量子回路モデルにおいては、必要な量子回路のゲート数、深さ、量子ビット数などで評価ができるが、以下では主にクエリ計算量 (query complexity) を用いる。クエリ計算量とは、特定のゲートまたは関数（オラクル、oracle）の呼び出し（クエリ）回数のことを指す。クエリ数による評価は、オラクルの内部についての構造を仮定することなく容易に計算量評価が可能でまた、他のアルゴリズムを組み合わせる際にも便利であるという利点がある。反面オラクル自体の構成に例えば量子ビットに関し指数時間のゲート操作を含む場合は、仮にクエリ計算量において多項式時間の量子加速が起きたとしても、依然として指数時間の計算量に関しての問題は解決されない。

ここではクエリ計算量を用いて最適化に関連する効率的な量子アルゴリズムを紹介する。具体的には量子振幅増幅アルゴリズムとその考案の元となった Grover 探索アルゴリズム、そして Grover 探索の応用である最小値探索量子アルゴリズムを紹介する。

2.4.1 量子振幅増幅アルゴリズム

量子振幅増幅（Quantum Amplitude Amplification) アルゴリズム [30] とは、量子状態の特定の成分（基底）のみの振幅を増大させる量子アルゴリズムである。初めにその適用範囲の広さから量子振幅増幅アルゴリズムを説明し、

Grover 探索をその一例として説明する。以下のアルゴリズムとそれに付随する説明は主に原論文から引用し適宜補足する。

古典計算においては、確率変数 x を持つ確率分布 $P(x)$ と x を引数としたブール値関数 χ が与えられていて、$\chi(x)=1$ となる結果が得られる確率が $a>0$ であった場合、平均的に $\frac{1}{a}$ 回の試行によって $\chi(x)=1$ を満たす結果が得られる。一方量子振幅増幅アルゴリズムを用いれば $\Theta\left(\frac{1}{\sqrt{a}}\right)$ の試行回数で条件を満たす状態が得られる。

$\mathcal{H}$ を量子系の状態を記述するヒルベルト空間とする。$\boldsymbol{\mathcal{A}}$ をヒルベルト空間 $\mathcal{H}$ における任意の観測を含まない量子アルゴリズム（またはユニタリ演算）として基底状態 $|0\rangle$ に作用して生じる状態が

$$|\Psi\rangle \equiv \boldsymbol{\mathcal{A}}|0\rangle \tag{2.4.1}$$

であるとする。$\mathcal{H}$ の正規直交基底を $\{|x\rangle\}_{x=1}^{\dim\mathcal{H}}$ として、任意のブール値関数 $\chi\colon \mathbb{Z}\to\{0,\ 1\}$ は $\mathcal{H}$ を二つの部分空間の直和

$$\mathcal{H} = \mathcal{H}_\chi \oplus \mathcal{H}_\chi^\perp \tag{2.4.2}$$
$$\mathcal{H}_\chi \equiv \mathrm{span}\{|x\rangle \in \mathcal{H}\ |\chi(x)=1\} \tag{2.4.3}$$
$$\mathcal{H}_\chi^\perp \equiv \mathrm{span}\{|x\rangle \in \mathcal{H}\ |\chi(x)=0\} \tag{2.4.4}$$

へ分解する。ここで $\mathcal{H}$ 上の純粋状態 $|\Psi\rangle$ は $\mathcal{H}_\chi$, $\mathcal{H}_\chi^\perp$ への射影演算子 P_χ, $P_\chi^\perp$[*1]を用いて

$$|\Psi\rangle = P_\chi|\Psi\rangle + P_\chi^\perp|\Psi\rangle = |\Psi_1\rangle + |\Psi_0\rangle \tag{2.4.5}$$

と 2 つの直交する成分 $|\Psi_1\rangle \equiv P_\chi|\Psi\rangle$, $|\Psi_0\rangle \equiv P_\chi^\perp|\Psi\rangle$ の和として書ける。

$|\Psi\rangle$ に対して $\mathcal{H}_\chi$ 上へ射影した状態 $|\Psi_1\rangle$ のノルム

$$a \equiv \langle\Psi_1|\Psi_1\rangle \tag{2.4.6}$$

は、測定オペレータ集合を $\{|x\rangle\langle x|\}_{x=1}^{\dim\mathcal{H}}$ とる射影測定をした際に観測される状態 $|x\rangle$ が $\mathcal{H}_\chi$ に属する確率、すなわち $\chi(x)=1$ を満たす確率となっている。

この振幅を増幅するための量子操作が

$$\mathbf{Q} \equiv \mathbf{Q}(\boldsymbol{\mathcal{A}},\chi) \equiv -\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}\mathbf{S}_\chi \tag{2.4.7}$$

であって、$\mathbf{Q}$ を状態 $|\Psi\rangle$ に繰り返し作用することで $\chi(x)=1$ を満たす状態の振幅を増幅することができる。ここでいう $\mathbf{S}_\chi$ とは条件を満たす状態の位相のみを反転させる量子操作であり、

$$|x\rangle \mapsto \begin{cases} -|x\rangle & (\chi(x)=1) \\ |x\rangle & (\chi(x)=0) \end{cases} \tag{2.4.8}$$

と正規直交基底に含まれる $|x\rangle$ を変換する。つまり $\mathbf{S}_\chi = \sum_x (-1)^{\chi(x)}|x\rangle\langle x|$。また $\mathbf{S}_0$ とは状態 $|0\rangle$ 成分のみの位相を反転させる量子操作であって

$$\mathbf{S}_0 \equiv I - 2|0\rangle\langle 0| \tag{2.4.9}$$

と表せる。$\mathbf{Q}$ が実際に条件を満たす状態を増幅させる量子操作であることを示すために、以下証明を行う。

Q を一回作用させた状態に関して次の補題 2.4.1 が成立する[*2]。

[*1] 射影演算子 P_χ, $P_\chi^\perp$ はエルミートで性質 $P_\chi = P_\chi^\dagger = P_\chi^2, P_\chi^\perp = P_\chi^{\perp\dagger} = P_\chi^{\perp 2}, P_\chi + P_\chi^\perp = 1$ を満たす。

[*2] $|\Psi\rangle = \boldsymbol{\mathcal{A}}|0\rangle$ を用いて

$$\mathbf{Q}|\Psi_1\rangle = -\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}\mathbf{S}_\chi|\Psi_1\rangle = (2|\Psi\rangle\langle\Psi| - 1)(-|\Psi_1\rangle) = (1-2a)|\Psi_1\rangle - 2a|\Psi_0\rangle$$
$$\mathbf{Q}|\Psi_0\rangle = -\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}\mathbf{S}_\chi|\Psi_0\rangle = (2|\Psi\rangle\langle\Psi| - 1)(+|\Psi_0\rangle) = 2(1-a)|\Psi_1\rangle + (1-2a)|\Psi_0\rangle$$

補題 2.4.1.

$$\mathbf{Q}\,|\Psi_1\rangle = (1-2a)\,|\Psi_1\rangle - 2a\,|\Psi_0\rangle \tag{2.4.10}$$
$$\mathbf{Q}\,|\Psi_0\rangle = 2(1-a)\,|\Psi_1\rangle + (1-2a)\,|\Psi_0\rangle \tag{2.4.11}$$

補題 2.4.1 から

$$\mathcal{H}_\Psi = \mathrm{span}\{|\Psi_1\rangle, |\Psi_0\rangle\} \tag{2.4.12}$$

とすると、部分空間 $\mathcal{H}_\Psi$ は $\mathbf{Q}$ の作用のもとで閉じている、つまり

$$|\phi\rangle \in \mathcal{H}_\Psi \Rightarrow \mathbf{Q}\,|\phi\rangle \in \mathcal{H}_\Psi \tag{2.4.13}$$

である。また

$$\dim\mathcal{H}_\Psi = \begin{cases} 2 & (0<a<1) \\ 1 & (a=0,1) \end{cases} \tag{2.4.14}$$

が成り立つ。補題 2.4.1 の帰結として次が成り立つ。

補題 2.4.2. $0<a<1$ とする。$\mathcal{H}_\Psi$ における $\mathbf{Q}$ の作用は以下の反転を行う演算子の積

$$U_\Psi U_{\Psi_0} \tag{2.4.15}$$

として表せる[*3]。ここで

$$U_{\Psi_0} \equiv I - \frac{2}{1-a}\,|\Psi_0\rangle\langle\Psi_0| \tag{2.4.16}$$
$$U_\Psi \equiv I - 2\,|\Psi\rangle\langle\Psi| \tag{2.4.17}$$

とした。演算子 U_{Ψ_0} は $|\Psi_0\rangle$ に関する反転を行い、演算子 U_Ψ は $|\Psi\rangle$ に関する反転を行う。（図 *2.8* 参照）

$\mathbf{Q}$ の作用に関する補足として、$\mathcal{H}$ における $\mathcal{H}_\Psi$ の直交補空間 $\mathcal{H}_\Psi^\perp$ 上で $\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}$ は恒等演算子 I に一致する[*4]から $\mathcal{H}_\Psi^\perp$ で $\mathbf{Q}$ は $-\mathbf{S}_\chi$ と一致し、$\mathbf{Q}^2$ は恒等演算子 I に一致する。従って $\mathcal{H}_\Psi^\perp$ 内で $\mathbf{Q}$ の全ての固有ベクトルは固有値 ±1 を持ち、$\mathcal{H}_\Psi^\perp$ 上での $\mathbf{Q}$ の作用が明らかになった。任意の $|\psi\rangle \in \mathcal{H}$ に対する $\mathbf{Q}$ の作用を調べるには $\mathcal{H}_\Psi$ への $|\psi\rangle$ の射影に対する $\mathbf{Q}$ の作用を調べれば十分であることがわかる。

以下しばらく $0<a<1$ として続ける。$\mathbf{Q}$ に関して閉じた $\mathcal{H}_\Psi$ は、$\mathbf{Q}$ のユニタリー性と式 (2.4.14) から以下の $\mathbf{Q}$ の 2 つの固有ベクトル

$$|\Psi_\pm\rangle \equiv \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{a}}\,|\Psi_1\rangle \pm \frac{i}{\sqrt{1-a}}\,|\Psi_0\rangle\right)\ (0<a<1) \tag{2.4.18}$$

でその正規直交基底が構成でき、満たす性質を補題 2.4.3 にまとめる。

補題 2.4.3. $|\Psi_\pm\rangle$ に対応する $\mathbf{Q}$ の固有値は

$$\lambda_\pm = e^{\pm 2i\theta_a} \tag{2.4.19}$$

[*3] $\mathcal{H}_\Psi$ を張る $|\Psi_1\rangle, |\Psi_0\rangle$ に実際に $U_\Psi U_{\Psi_0}$ を施し補題 2.4.1 を用いて確認できる。

$$U_\Psi U_{\Psi_0}\,|\Psi_1\rangle = (I - 2\,|\Psi\rangle\langle\Psi|)\,|\Psi_1\rangle = 2(1-a)\,|\Psi_1\rangle + (1-2a)\,|\Psi_0\rangle = Q\,|\Psi_1\rangle$$
$$U_\Psi U_{\Psi_0}\,|\Psi_0\rangle = U_\Psi\left(I - \frac{2}{1-a}\,|\Psi_0\rangle\langle\Psi_0|\right)|\Psi_0\rangle = -(I - 2\,|\Psi\rangle\langle\Psi|)\,|\Psi_0\rangle = 2(1-a)\,|\Psi_1\rangle + (1-2a)\,|\Psi_0\rangle = Q\,|\Psi_0\rangle$$

[*4] 任意の $|\psi\rangle \in \mathcal{H}_\Psi^\perp$ に対し $\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}\,|\psi\rangle = (I - 2\,|\Psi\rangle\langle\Psi|)\,|\psi\rangle = |\psi\rangle$

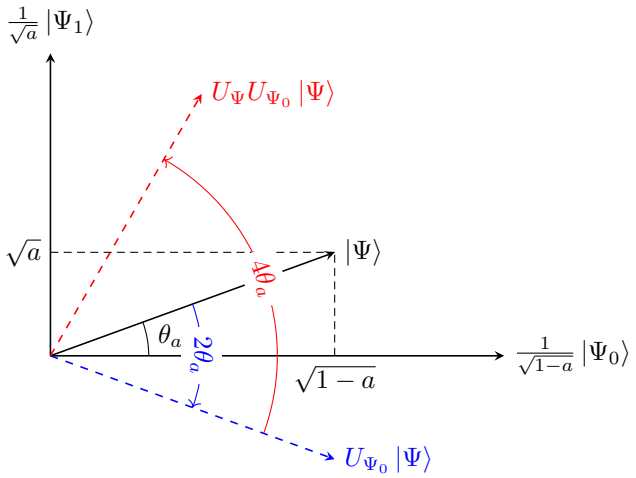


図 2.8: 振幅増幅の様子

増幅させる状態 $|\Psi\rangle$ を解の成分 $|\Psi_1\rangle$ とそうでない成分 $|\Psi_0\rangle$ に分解し、それら 2 つを軸とした二次元平面内で表現する。U_{Ψ_0} が $|\Psi_0\rangle$ に対する反転を行い、U_Ψ が $|\Psi_0\rangle$ に対する反転を行う。$U_\Psi U_{\Psi_0}$ を繰り返すことで解の成分 $|\Psi_1\rangle$ が増大する。

である[*5]。ここで θ_a は以下を満たす。

$$\sin^2(\theta_a) \equiv a = \langle\Psi_1|\Psi_1\rangle \tag{2.4.20}$$

$$0 \leq \theta_a \leq \pi/2 \tag{2.4.21}$$

$|\Psi\rangle$ を $\mathcal{H}_\Psi$ の基底で書き直せば、$\mathbf{Q}$ が実際に増幅する過程であることが以下のように見てとれる。

$$|\Psi\rangle = \frac{-i}{\sqrt{2}}\left(e^{i\theta_a}|\Psi_+\rangle - e^{-i\theta_a}|\Psi_-\rangle\right) \tag{2.4.22}$$

$$\begin{aligned}\therefore \mathbf{Q}^j|\Psi\rangle &= \frac{-i}{\sqrt{2}}\left(e^{(2j+1)i\theta_a}|\Psi_+\rangle - e^{-(2j+1)i\theta_a}|\Psi_-\rangle\right)\\ &= \frac{1}{\sqrt{a}}\sin\left((2j+1)\theta_a\right)|\Psi_1\rangle + \frac{1}{\sqrt{1-a}}\cos\left((2j+1)\theta_a\right)|\Psi_0\rangle\end{aligned} \tag{2.4.23}$$

よって $0 < a < 1$ においてある非負整数 m に対し $\mathbf{Q}^m\boldsymbol{\mathcal{A}}|0\rangle$ を射影測定して $\chi(x) = 1$ を満たす状態 $|x\rangle$ が得られる確率は、$\sin^2\left((2m+1)\theta_a\right)$。この結果は $a = 0, 1$ であっても成り立つ。

従ってアルゴリズム $\boldsymbol{\mathcal{A}}$ の成功確率を増大させるには、$\sin^2\left((2m+1)\theta_a\right)$ が 1 に近づくような非負整数 m を選べばいい。そのような m の選び方は次の定理が与える。

定理 2.4.1. 量子振幅増幅：a が既知な場合での 2 次の高速化

$\boldsymbol{\mathcal{A}}$ を任意の測定を含まない量子アルゴリズムまたはユニタリ演算子、$\chi\colon \mathbb{Z} \to \{0,1\}$ を任意のブール値関数と

[*5] 式 (2.4.16)、(2.4.17) から

$$\begin{aligned}\mathbf{Q}|\Psi_\pm\rangle &= \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{a}}\left\{(1-2a)|\Psi_1\rangle - 2a|\Psi_0\rangle\right\} \pm \frac{i}{\sqrt{1-a}}\left\{2(1-a)|\Psi_1\rangle + (1-2a)|\Psi_0\rangle\right\}\right]\\ &= \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{a}}\left(1-2a \pm 2i\sqrt{a(1-a)}\right)|\Psi_1\rangle \pm \frac{1}{\sqrt{1-a}}\left(-2\sqrt{a(1-a)} \pm i(1-2a)\right)|\Psi_0\rangle\right]\\ &= \left(\sqrt{1-a} \pm i\sqrt{a}\right)^2|\Psi_\pm\rangle = e^{\pm 2i\theta_a}|\Psi_\pm\rangle\end{aligned}$$

する。a を $\boldsymbol{\mathcal{A}}$ の成功する確率とする。$a > 0$ と仮定して、$m = \lfloor \pi/4\theta_a \rfloor$ とおく。ただし θ_a は $0 < \theta_a \leq \pi/2$ かつ $\sin^2(\theta_a) = a$ なるものとする。

このとき、状態 $\mathbf{Q}^m\boldsymbol{\mathcal{A}}|0\rangle$ を作り系を $\mathcal{H}$ の正規直交基底 $\{|x\rangle\langle x|\}_{x=1}^{dim\mathcal{H}}$ で射影測定することで条件 $\chi(x) = 1$ を満たす状態 $|x\rangle$ が少なくとも $\max(1-a, a)$ の確率で得られる[*6]。

上記の定理によって 2 次の高速化が可能であるのは、$\chi(x) = 1$ を満たす状態が、高々期待値 $(2m+1)/\max(1-a,a) = \Theta(1/\sqrt{a})$[*7]回の $\boldsymbol{\mathcal{A}}$ と $\boldsymbol{\mathcal{A}}^{-1}$ の試行で得られるからである。ただし注意点があり、この定理の主張するところは状態 $\mathbf{Q}^m\boldsymbol{\mathcal{A}}|0\rangle$ を得るにはあらかじめ m つまり a の値が知られていなければならない点である。a の値が未知の場合には定理 2.4.2 が存在するが、その前に量子振幅増幅アルゴリズムについて補足を加えておく。

量子振幅増幅アルゴリズムの適用先として考えられるのは、自然数 N としてブール値関数 $f\colon \{0,1,\ldots,N-1\} \to \{0,1\}$ が与えられており、$f(x) = 1$ を満たす解 x_0 がただ一つ存在していると仮定して、x_0 を探し出す探索問題である。f の構造が不明すなわち f が入力に対する関数の出力のみしか知り得ないブラックボックスである場合、古典的にはしらみ潰しに行う必要があり、探索空間のサイズ N のオーダー $\Theta(N)$ の関数の呼び出し（クエリ）が必要になる。

一方 Grover は量子計算によって上述の探索問題が $O(\sqrt{N})$ の呼び出しで十分であることを示した。その手続きは Grover 探索アルゴリズム [31, 32, 33] と呼ばれ、量子振幅増幅アルゴリズムにおいて $\chi = f, a = 1/\sqrt{N}$ とし n 量子ビットに対するアダマール変換 $\boldsymbol{\mathcal{A}} = H^{\otimes n}$ として始状態を一様な重みの状態

$$|\Psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{2.4.24}$$

に選んだ場合と一致する。探索の量子アルゴリズムの下限が $\Omega(\sqrt{N})$ であることが示されている [34] ため、Grover のアルゴリズムはオーダーとして最良であることが知られている。

量子振幅増幅アルゴリズムを実際の問題に適用する際に用いる際次の補題 2.4.4 が成り立つ。

補題 2.4.4. 量子アルゴリズム $\boldsymbol{\mathcal{A}}$ が 2 回のユニタリ変換の積

$$\boldsymbol{\mathcal{A}} \equiv \mathcal{A}_{table}\left(\mathcal{A}_{init} \otimes I_{2^n}\right) \tag{2.4.25}$$

として書けてそれぞれ状態 $|0\rangle_1, |i\rangle_1|0\rangle_2$ に対する作用が

$$\mathcal{A}_{init}|0\rangle_1 = \sum_{i \in G_{\boldsymbol{\mathcal{A}}}} a_i |i\rangle_1 \tag{2.4.26}$$

$$\mathcal{A}_{table}|i\rangle_1|0\rangle_2 = |i\rangle_1|A(i)\rangle_2 \tag{2.4.27}$$

[*6] $\pi/4\theta_a - 1 < \lfloor \pi/4\theta_a \rfloor \leq \pi/4\theta_a$ から $-\theta_a < (2m+1)\theta_a - \frac{\pi}{2} \leq \theta_a$ なので

$$\sin^2\left((2m+1)\theta_a\right) = \cos^2\left((2m+1)\theta_a - \frac{\pi}{2}\right) \geq \cos^2(\theta_a) = 1 - a$$

$\pi/4 < \theta_a \leq \pi/2$ の時、$m = \lfloor \pi/4\theta_a \rfloor = 0$ より $\sin^2\left((2m+1)\theta_a\right) = \sin^2(\theta_a) = a$

$$\therefore \sin^2\left((2m+1)\theta_a\right) \geq \max(1-a, a)$$

[*7] $2m+1 \begin{cases} \leq 2(\pi/2\theta_a + 1) = 2(\pi/2\arcsin(\sqrt{a}) + 1) \leq 2(\pi/2\sqrt{a} + 1) \\ > (\pi/2\theta_a - 1) = (\pi/2\arcsin(\sqrt{a}) - 1) \geq (1/\sqrt{a} - 1) \end{cases}$ と $1/2 \leq \max(1-a, a) \leq 1$ から

$$(2m+1)/\max(1-a,a) = \Theta(1/\sqrt{a})$$

である場合を考える。ただし $n, m \in \mathbb{N}, \sum_{i \in G_{\boldsymbol{\mathcal{A}}}} |a_i|^2 = 1, A(i) \in \mathbb{Z}_{2^n}$。これに合わせてブール値関数 χ の定義である式 (2.4.8) を

$$|x\rangle_2 \mapsto \begin{cases} -|x\rangle_2 & (\chi(x) = 1) \\ |x\rangle_2 & (\chi(x) = 0) \end{cases} \tag{2.4.28}$$

と変更しこれを $\boldsymbol{S}_\chi \equiv I_{2^m} \otimes \sum_x (-1)^{\chi(x)} |x\rangle_{22}\langle x|$ とする。

この時 $\boldsymbol{S}_0$ は以下の $\boldsymbol{S}_0'$ で十分である[*8]。

$$\boldsymbol{S}_0' = (I_{2^m} - 2|0\rangle_{11}\langle 0|) \otimes I_{2^n} \tag{2.4.29}$$

条件を満たせば補題 2.4.4 によって $\boldsymbol{S}_0$ をより少ないゲート数で構築することができる。第 3 章でこの補題 2.4.4 を用いて $\boldsymbol{S}_0'$ を構築する。

a の値が知られているという仮定で二次の高速化が可能であることを定理 2.4.1 で見たが、以下の定理では一般に未知であっても二次の高速化が可能であることを示す。

定理 2.4.2. 量子振幅増幅：a が未知な場合での二次の高速化

以下を満たす **QSearch** と呼ばれる量子アルゴリズムが存在する。

$\boldsymbol{\mathcal{A}}$ を任意の測定を含まない量子アルゴリズムまたはユニタリ演算子、$\chi\colon \mathbb{Z} \to \{0, 1\}$ を任意のブール値関数とする。a を $\boldsymbol{\mathcal{A}}$ の成功する確率とする。

$a > 0$ ならば、アルゴリズム **QSearch** を用いることで $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}}^{-1}$ を期待値 $\Theta\left(1/\sqrt{a}\right)$ 回行うことで $\chi(x) = 1$ を満たす解 $|x\rangle$ が得られる。$a = 0$ ならば **QSearch** は停止しない。

証明の概略を示す。$a \geq 3/4$ の時、**QSearch**$(\boldsymbol{\mathcal{A}}, \chi)$ 内のステップ 3 の存在が、解が早期に得られることを保証している。よって $0 < a < 3/4$ の場合の計算量を調べればいい。そのために補題 2.4.5 を用いる。これは帰納的に加法定理と 2 倍角の公式を用いれば示せる。

補題 2.4.5. $m \in \mathbb{N}, \alpha \in \mathbb{R}$ とする。

$$\sum_{j=0}^{m-1} \cos((2j+1)\alpha) = \frac{\sin(2m\alpha)}{2\sin\alpha} \tag{2.4.31}$$

補題 2.4.5 から $0 < a < 3/4$ で **QSearch**$(\boldsymbol{\mathcal{A}}, \chi)$ 内のステップ 2 から 7 において $\chi(z) = 1$ なる状態 $|z\rangle$ が得られる

[*8] 補題の成立を確認する。まず

$$\boldsymbol{\mathcal{A}}|0\rangle_1|0\rangle_2 = \sum_{i \in G_{\boldsymbol{\mathcal{A}}}} a_i |i\rangle_1 |A(i)\rangle_2 \tag{2.4.30}$$

が成り立つ。そして状態 $\boldsymbol{\mathcal{A}}|0\rangle_1|0\rangle_2$ に増幅 $\mathbf{Q}$ を一回行った状態を比較すると、

$$\begin{aligned}
\boldsymbol{\mathcal{A}}\boldsymbol{S}_0'\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{S}_\chi\boldsymbol{\mathcal{A}}|0\rangle_1|0\rangle_2 &= \mathcal{A}_{\text{table}} \left(\left(I_{2^m} - 2 \sum_{i,j \in G_{\boldsymbol{\mathcal{A}}}} a_i a_j^* |i\rangle_{11}\langle j| \right) \otimes I_{2^n} \right) \sum_{k \in G_{\boldsymbol{\mathcal{A}}}} a_k (-1)^{\chi(A(k))} |k\rangle_1 |0\rangle_2 \\
&= \sum_{i,j \in G_{\boldsymbol{\mathcal{A}}}} a_i |a_j|^2 (-1)^{\chi(A(j))} |j\rangle_1 |A(j)\rangle_2 \\
\boldsymbol{\mathcal{A}}\boldsymbol{S}_0\boldsymbol{\mathcal{A}}^{-1}\boldsymbol{S}_\chi\boldsymbol{\mathcal{A}}|0\rangle_1|0\rangle_2 &= \mathcal{A}_{\text{table}} \left(I_{2^{n+m}} - 2 \sum_{i,j \in G_{\boldsymbol{\mathcal{A}}}} a_i a_j^* |i\rangle_{11}\langle j| \otimes |0\rangle_{22}\langle 0| \right) \sum_{k \in G_{\boldsymbol{\mathcal{A}}}} a_k (-1)^{\chi(A(k))} |k\rangle_1 |0\rangle_2 \\
&= \sum_{i,j \in G_{\boldsymbol{\mathcal{A}}}} a_i |a_j|^2 (-1)^{\chi(A(j))} |j\rangle_1 |A(j)\rangle_2
\end{aligned}$$

と両者が一致することが具体的な計算によって確認でき、$\mathbf{Q}$ を複数回作用させた場合でも同様である。

Algorithm 1 QSearch$(\boldsymbol{\mathcal{A}}, \chi)$

1: $l = 0$ とおき、c を $1 < c < 2$ を満たすある定数とする。
2: $l \leftarrow l + 1$ とし、$M = \lceil c^l \rceil$ とする。
3: 始状態 $|0\rangle$ に量子アルゴリズム $\boldsymbol{\mathcal{A}}$ を作用させ系を測定し測定結果 $|z\rangle$ を得る。
4: **if** $\chi(z) = 1$ **then return** z
5: **else**
6: 系を状態 $\boldsymbol{\mathcal{A}}|0\rangle$ に初期化する。
7: 整数 j を 1 から M までの整数から一様ランダムに選ぶ。
8: $\mathbf{Q} = \mathbf{Q}(\boldsymbol{\mathcal{A}}, \chi)$ としてレジスタに $\mathbf{Q}^j$ を施し系を測定し測定結果 $|z\rangle$ を得る。
9: **if** $\chi(z) = 1$ **then return** z
10: **else**
11: **go to** step 2
12: **end if**
13: **end if**

確率は、

$$a + (1-a)\frac{1}{M}\sum_{j=1}^{M}\sin^2\left((2j+1)\theta_a\right) \geq \frac{1}{2}\left(1 - \frac{1}{2M\sqrt{a}}\right) \tag{2.4.32}$$

と下から抑えられる[*9]。これは $a \geq 3/4$ でも成り立つ。

定数 c_0, M_0 を

$$c_0 \equiv 1 - \frac{1}{2}c \tag{2.4.33}$$

$$M_0 \equiv \frac{1}{2c_0\sqrt{a}} \tag{2.4.34}$$

とおく。T_1 を $M < M_0$ における $\boldsymbol{\mathcal{A}}$ の最大の適用回数、T_2 を $M \geq M_0$ における $\boldsymbol{\mathcal{A}}$ の適用回数の期待値とすると、解が得られるまでの **QSearch**$(\boldsymbol{\mathcal{A}}, \chi)$ 全体における $\boldsymbol{\mathcal{A}}$ の適用回数の期待値は明らかに $T_1 + T_2$ で抑えることができる。よって T_1, T_2 の $\boldsymbol{\mathcal{A}}$ の適用回数に対する振る舞いを調べればいい。

T_1 については

$$\begin{aligned} T_1 = \sum_{l=1}^{\lfloor \log M_0 \rfloor}\left(1 + (\lceil c^l \rceil + 1)\right) \leq \sum_{l=1}^{\lfloor \log M_0 \rfloor}\left(3 + c^l\right) = 3\lfloor \log M_0 \rfloor + c\frac{c^{\log M_0} - 1}{c-1} \\ = 3\lfloor \log M_0 \rfloor + c\frac{M_0^{\log c} - 1}{c-1} = O(M_0) = O\left(1/\sqrt{a}\right) \ (\because \log c < 1) \end{aligned} \tag{2.4.35}$$

[*9]

$$\begin{aligned} \frac{1}{M}\sum_{j=1}^{M}\sin^2\left((2j+1)\theta_a\right) &= \frac{1}{2M}\sum_{j=1}^{M}\left(1 - \cos\left(2\theta_a(2j+1)\right)\right) = \frac{1}{2} - \frac{1}{2M}\left(\frac{\sin\left(4(M+1)\theta_a\right)}{2\sin\left(2\theta_a\right)} - \cos\left(2\theta_a\right)\right) \\ &\geq \frac{1}{2}\left(1 - \frac{1}{2M\sin\left(2\theta_a\right)} + \frac{1}{2M}\cos\left(2\theta_a\right)\right) = \frac{1}{2}\left(1 - \frac{1}{4M\sqrt{a(1-a)}} + \frac{1}{2M}(1-2a)\right) \\ &\geq \frac{1}{2}\left(1 - \frac{1}{2M\sqrt{a}} + \frac{1}{2M}(1-2a)\right) \end{aligned}$$

最後の式変形において $0 < a < 3/4$ を用いた。

$$\therefore a + (1-a)\frac{1}{M}\sum_{j=1}^{M}\sin^2\left((2j+1)\theta_a\right) \geq \frac{1}{2}\left(1 - \frac{1}{2M\sqrt{a}}\right) + \frac{(1-a)(1-2a)}{2M} + a\left(\frac{1}{2} + \frac{1}{2M\sqrt{a}}\right) \geq \frac{1}{2}\left(1 - \frac{1}{2M\sqrt{a}}\right)$$

と $T_1 = O(1/\sqrt{a})$ が言えた。

T_2 については式 (2.4.32) からステップ 9 における失敗確率が $p_0 \equiv (1+c_0)/2$ によって上から抑えることができる。よって $\lceil c^{i+1} \rceil \geq M_0 > \lceil c^i \rceil$ なる i を i_0 として、任意の $i > i_0$ に対し高々 $p_0^{i-i_0}$ の確率で $\lceil c^i \rceil$ なる M をステップ 2 で得る。

$$\therefore T_2 = O\left(\sum_{i>i_0} \left(1 + (1 + \lceil c^i \rceil)\right) p_0^{i-i_0}\right) = O\left(\sum_{i=1} \left(3 + M_0 c^i\right) p_0^i\right) = O(M_0) = O\left(1/\sqrt{a}\right) \tag{2.4.36}$$

途中で $cp_0 < 1$ を用いた。故に **QSearch**$(\boldsymbol{\mathcal{A}}, \chi)$ 全体における $\boldsymbol{\mathcal{A}}$ の適用回数の期待値は $O\left(1/\sqrt{a}\right)$。$\Omega\left(1/\sqrt{a}\right)$ でもあることの証明は省略する。$a = 0$ である時は j 回の振幅増幅後も解が得られる確率は常に $\sin^2\left((2j+1)\theta_a\right) = 0$ でありアルゴリズムは停止しない。

a が既知である場合にまた話が戻るが、その場合解を決定論的につまり確率 1 で $\Theta(1/\sqrt{a})$ の計算量により得ることができる。その前に $\mathbf{S}_\chi, \mathbf{S}_0$ を一般化する。$\mathbf{S}_\chi$ の正規直交基底に含まれる状態 $|x\rangle$ に対する作用が

$$|x\rangle \mapsto \begin{cases} e^{i\varphi} |x\rangle & (\chi(x) = 1) \\ |x\rangle & (\chi(x) = 0) \end{cases} \tag{2.4.37}$$

と、解の状態のみに位相 φ を加えるユニタリ変換とする。また $\mathbf{S}_0$ として基底状態 $|0\rangle$ の位相にのみ位相 ϕ を加える

$$\mathbf{S}_0(\phi) \equiv I - (1 - e^{i\phi}) |0\rangle\langle 0| \tag{2.4.38}$$

とする。これに応じて振幅増幅操作 $\mathbf{Q}(\boldsymbol{\mathcal{A}}, \chi)$ を $\mathbf{Q}(\boldsymbol{\mathcal{A}}, \chi, \varphi, \phi)$ とおき直せば、以下が同様に成り立つ。

補題 2.4.6.

$$\mathbf{Q} |\Psi_1\rangle = e^{i\varphi} \left(\left(1 - e^{i\phi}\right)\right) |\Psi_1\rangle + e^{i\varphi} \left(1 - e^{i\phi}\right) a |\Psi_0\rangle \tag{2.4.39}$$

$$\mathbf{Q} |\Psi_0\rangle = \left(1 - e^{i\phi}\right)(1 - a) |\Psi_1\rangle - \left(\left(1 - e^{i\phi}\right) a + e^{i\phi}\right) |\Psi_0\rangle \tag{2.4.40}$$

すると $\widetilde{m} = \pi/4\theta_a - 1/2$ として拡張前の振幅増幅 $\mathbf{Q}(\boldsymbol{\mathcal{A}}, \chi, \pi, \pi)$ を $\widetilde{m}$ 回行った時の状態は

$$\frac{1}{\sqrt{a}} \sin\left((2\lfloor \widetilde{m} \rfloor + 1)\theta_a\right) |\Psi_1\rangle + \frac{1}{\sqrt{1-a}} \cos\left((2\lfloor \widetilde{m} \rfloor + 1)\theta_a\right) \Psi_0 \tag{2.4.41}$$

である。その後以下

$$e^{i\varphi} \left(1 - e^{i\phi}\right) \sqrt{a} \sin\left((2\lfloor \widetilde{m} \rfloor + 1)\theta_a\right) = \left(\left(1 - e^{i\phi}\right) a + e^{i\phi}\right) \frac{1}{\sqrt{1-a}} \cos\left((2\lfloor \widetilde{m} \rfloor + 1)\theta_a\right) \tag{2.4.42}$$

を満たす φ, ϕ に対して振幅増幅 $\mathbf{Q}(\boldsymbol{\mathcal{A}}, \chi, \varphi, \phi)$ を一回行うと、解の成分の振幅が 0 になる。こうした φ, ϕ がとれることは上の条件式を変形した式

$$\cot\left((2\lfloor \widetilde{m} \rfloor + 1)\theta_a\right) = e^{i\varphi} \sin(2\theta_a) \left(-\cos(2\theta_a) + i \cot(\phi/2)\right)^{-1} \tag{2.4.43}$$

が必ず解を持つことから見てとれる。

ここまで説明してきたように、量子振幅増幅アルゴリズムを探索問題に適用することができて、古典的なしらみつぶし (brute force) 探索に比べて 2 次の高速化が得られることを示してきた。しかし、多くの問題において探索問題には構造が存在しており、古典計算で適切な発見的手法（ヒューリスティックス）を用いればしらみつぶし（全）探索よりも早く探索が可能である。こうした場合にも量子振幅増幅アルゴリズムは応用することができ、以下の限定的なヒューリスティックス[*10]に対しても 2 次の高速化が実現できることを見ていく。

[*10] 必ずしも全てのヒューリスティックスがこの定式化に当てはまるとは限らない。元来は答えの精度が保証されない発見的な手法のことを広く一般に指す。

定義 2.4.1. 関数族 $\mathcal{F}$ を集合 X 上のブール値関数 $f\colon X \to \{0,1\}$ の集合とする。
適切な有限集合 R に対し関数 $G\colon \mathcal{F} \times R \to X$ のことをここでは暫定的に “ヒューリスティックス” と呼ぶ。

ヒューリスティックスを用いて与えられた関数 f に対し $f(x) = 1$ なる $x \in X$ を得ることを目指す。以下ヒューリスティックスを多項式時間で動作し、十分な確率で解を得る確率的なアルゴリズムを指すとする。ヒューリスティックス G は乱数のシード値 $r \in R$ を用い $f(x) = 1$ なる解 x の推測を行う。関数 $f \in \mathcal{F}$ に対して対応する解の個数を $t_f = |\{x \in X \mid f(x) = 1\}|$、解を与える乱数のシード値の個数を $h_f = |\{r \in R \mid f(G(f,r)) = 1\}|$ とする。

与えられた関数 f に対し ${}^{h_f}/_{|R|} > {}^{t_f}/_{|X|}$ の場合に、ヒューリスティックスが “効率的” という。これはすなわち、乱数のシード値と f を G に入力し生成した解の候補は、一様ランダムに解の候補を生成する場合よりも、より高い確率で $f(x) = 1$ なる正しい解 x が得られることを意味している。これを拡張し複数の関数に関してヒューリスティックスが有効であるかを考える。$\forall f \in \mathcal{F}$ に対して f を $\mathcal{F}$ 上の確率分布に基づき確率的に選択し、乱数のシード値と G を用いて生成した解の候補の方が、一様ランダムに解の候補を推測する場合よりも期待値として高い確率

$$E_{\mathcal{F}}\left(\frac{h_f}{|R|}\right) > E_{\mathcal{F}}\left(\frac{t_f}{|X|}\right) \tag{2.4.44}$$

で正しい解が得られる場合に、ヒューリスティックス G を “良い” とする。こうして定義したヒューリスティックスが量子振幅により二次の高速化が実現されることを次の定理が保証する。

定理 2.4.3. ヒューリスティックスにおける二次の高速化
ブール値関数の族 $\mathcal{F}$ を $\mathcal{F} \subseteq \{f \mid X \to \{0,1\}\}$、$\mathcal{D}$ を $\mathcal{F}$ 上で定義された確率分布とする。
古典コンピューター上で確率分布 $\mathcal{D}$ に従って f を取り出し、ヒューリスティックス $G\colon \mathcal{F} \times R \to X$ に f と乱数のシード値を与え $f(x_0) = 1$ なる $x_0 \in X$ をある確率で期待値 $O(T)$ の実行時間で得ることができる時、量子コンピューターを用いて同じことが期待値 $O(\sqrt{T})$ の実行時間で可能である。

証明. 古典的ヒューリスティックス G を量子アルゴリズム **QSearch** に入力すればいい。

$\boldsymbol{\mathcal{A}}$ として一様なシード値 $r \in R$ の重ね合わせ状態を作る変換を考え、任意の関数 $f \in \mathcal{F}$ に対し $r \in R$, $\chi(r) = f(G(f,r))$ として量子振幅増幅により f に対し G を用いて正しい解を与えるようなシード値を増幅する。$x = G\left(f, \mathbf{QSearch}(\boldsymbol{\mathcal{A}}, \chi)\right)$ として解の候補を与えると、定理 2.4.2 によって、ヒューリスティック探索の実行時間の期待値が $\Theta\left(\sqrt{\frac{|R|}{h_f}}\right)$ になる。P_f を関数 f が選択される確率とすると $\sum_{f\in\mathcal{F}} P_f = 1$ であって、G を用いた実行時間の $\mathcal{F}$ 上の期待値が $\Theta\left(\sum_{f\in\mathcal{F}} \sqrt{\frac{|R|}{h_f}} P_f\right)$ となる。これを書き換えるとコーシー・シュワルツの不等式から

$$\sum_{f\in\mathcal{F}} \sqrt{\frac{|R|}{h_f}} P_f \sqrt{P_f} \leq \sqrt{\sum_{f\in\mathcal{F}} \frac{|R|}{h_f} P_f} \sqrt{\sum_{f\in\mathcal{F}} P_f} = \sqrt{\sum_{f\in\mathcal{F}} \frac{|R|}{h_f} P_f} \tag{2.4.45}$$

となる。 □

定理 2.4.3 の証明を、ヒューリスティックス G 自体を可逆な形式で量子コンピューター内で構築することで行うこともできるが上述の証明と等価であるため省く。この定理はヒューリスティックス探索を用いることのできる “情報のある探索” の場合でも、Grover 増幅ひいては量子振幅増幅が有用であることを示している。ここでのヒューリスティックスを拡張し第 5 章で議論を行う。

原論文 [30] においてはさらに位相推定アルゴリズム [35, 36] と組み合わせることで解の数を推定する量子計数（Quantum Counting）や、解に対応する状態の振幅を推定する量子振幅推定（Quantum Amplitude Estimation）が可能であることを示している。また最近の研究成果として観測結果に古典的最尤法を用いることで位相推定アルゴリズムなしに量子振幅推定が可能であることが知られている [37]。

2.4.2 最小値探索量子アルゴリズム

ここでは最適化問題に対する Grover 探索の応用を見ていく。（最適化の問題設定と比較対象となる古典アルゴリズムに関しては付録 C を参照。）Grover 増幅の一般的形式である量子振幅増幅アルゴリズムを介して、Grover 探索は brute force な探索に比べて二次の高速化が可能であることは前節で見た。証明の際に増幅する解に対応する状態の振幅が未知である時の対処を述べたが、本質的には時系列的に Boyer らによって解決され [38]、BBHT アルゴリズムと呼ばれる。Boyer は同時に探索空間のサイズ N に対し Grover 増幅の解が複数（$M > 1$ 個）存在する場合のクエリ計算量が $O(\sqrt{N/M})$ を与えている。その際パラメータ λ は $1 < \lambda < 4/3$ であれば良いと示した。

BBHT アルゴリズムを利用し Dürr と Høyer はインデックスとそれに対応する表（つまり離散集合上の関数）が存在しているとき、$O(\sqrt{N})$ のクエリで少なくとも $1/2$ より大きな確率で、表の値が最小となるインデックスを取り出すことが可能であることを示した [39]。それによれば $22.5\sqrt{N} + \log^2 N$ のクエリ回数に達した時をアルゴリズムの停止条件とし、BBHT における増幅条件をアルゴリズムの各ステップで得られている最小値の候補を閾値として設定するもので、最小値探索量子アルゴリズムまたは DH アルゴリズムと呼ばれる。（**Algorithm 2** 参照）同アルゴリズムは構造を仮定しない最適化問題へ適用可能であり、二次の加速をもたらす。証明はのちの第 5 章での議論と重複する部分があるため省略する。

Algorithm 2 Quantum algorithm for finding the minimum, **DH** Algorithm

Initialize:

$m = 1$

$\lambda = 8/7$ とおく。

$x_1 \in S$ を S 上一様ランダムに選択し、$y_1 = f(x_1)$ とする

for $i = 1, 2, \ldots$ *until a termination condition is met* **do**

0 以上 m 未満の整数から一様ランダムに整数 r_n を取り出す。

条件 $f(x) < y_n$ を満たすものを解 $|x\rangle$ として r_n 回の Grover 増幅を行い、観測後の解の候補を x、対応する関数値 $y = f(x)$ とおく。

if $y < y_n$ **then**

$x_{n+1} = x, y_{n+1} = y, m = 1$

else

$x_{n+1} = x_n, y_{n+1} = y_n, m \leftarrow \lambda m$

end if

end for

また DH アルゴリズムは、Baritompa ら (BBW) により考案された Grover Adaptive Search(GAS) アルゴリズム（**Algorithm 3** 参照）の一種とされる。実際初期化 $m = 1$ と、GAS における Grover 回転角 $\{r_n\}$ を BBHT と同様に $0 \le r_n < m$ から一様ランダムかつ $m \leftarrow \lambda m$ (if $y \ge y_n$), 1 (otherwise) とすれば一致する。命名の由来は古典的な確率的アルゴリズムの Pure Adaptive Search における閾値条件を Grover 増幅の条件としたもので、二次の加速が生まれる。Baritompa らは DH アルゴリズムを解析し λ の最良の値として $\lambda \cong 1.34$ を与えている。

しかし前節で議論したように一般に多くの問題では一定の構造が存在しそれにあったアルゴリズムや解の精度が保証されないまでも、経験的に比較的少ない計算量で解を得ることのできる発見的手法が存在する。これに対応する一つの解決策は前節のヒューリスティックスを Grover 探索に含めることであった。第 5 章において局所的構造が存在している場合に関して確率的局所最適化を組み合わせた場合の一般的な量子加速について述べる。

Algorithm 3 Grover Adaptive Search, **GAS**($\{r_i\}$)

$x_1 \in S$ を S 上一様ランダムに選択し、$y_1 = f(x_1)$ とする
for $i = 1, 2, \ldots$ *until a termination condition is met* **do**
  条件 $f(x) < y_n$ を満たすものを解 $|x\rangle$ として r_n 回の Grover 増幅を行い、出力を x, y とおく。
  if $y < y_n$ **then**
    $x_{n+1} = x, y_{n+1} = y$
  else
    $x_{n+1} = x_n, y_{n+1} = y_n$
  end if
end for

2.4.3 Jordan の量子勾配推定アルゴリズム

最適化問題において勾配情報を用いることが可能な時、そうでない場合に比べ最適化が容易になる。しかし勾配を計算するには、問題が複雑になればなるほど困難になる。

その事実を確認するために勾配の計算量をクエリ計算量によって評価する。まず実数値関数 $f\colon \mathbb{R}^d \to \mathbb{R}$ を定める。定義により座標 $\boldsymbol{x}$ における勾配 ∇f の各成分 $\frac{\partial f}{\partial x_i} (i = 1, \ldots, d)$ は、微小量 δ を用いて

$$\frac{\partial f}{\partial x_i} = \frac{f(\boldsymbol{x} + \delta \boldsymbol{e_i}) - f(\boldsymbol{x})}{\delta} + O(\delta) \tag{2.4.46}$$

と表せる。ここで $\{\boldsymbol{e}_i\}_{i=1}^d$ を $\mathbb{R}^d$ の標準基底とした。この式から勾配計算のために関数 f は少なくとも異なる座標 $\boldsymbol{x}, \boldsymbol{x} + \delta \boldsymbol{e}_1, \ldots, \boldsymbol{x} + \delta \boldsymbol{e}_d$ において計 $d+1$ 回評価されている（図 2.9 参照）ため、古典的クエリ計算量は $\Omega(d)$ である。

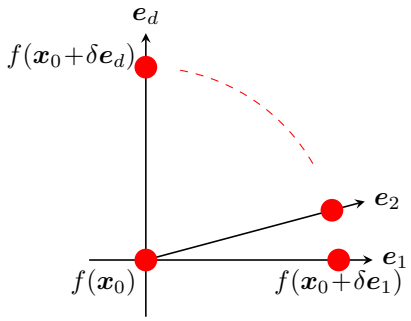


図 2.9: 古典的に勾配を計算するために最低限必要な座標点

一方 Jordan の量子アルゴリズム [40] を用いることで関数に対する線形性の仮定の上でクエリ計算量 $O(1)$ つまり定数で勾配を推定することができる。具体的にはブラックボックスな d 個の実変数を持つ実数値関数 $f : \mathbb{R}^d \to \mathbb{R}$ が、ある空間領域で微小量 $\|\delta\|$ により

$$f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x}) + \nabla f \cdot \boldsymbol{\delta} + O\left(\|\boldsymbol{\delta}\|^2\right) \tag{2.4.47}$$

と表される場合、$\mathcal{O}(1)$ のクエリで勾配 ∇f が得られる。ここでいうブラックボックスとは関数の入力に対し出力のみしか知り得ないという関数の性質を意味する。その具体的手続きを一部簡素化して述べると、基底状態 $|\boldsymbol{0}\rangle$ にアダマール変換を行い、$\mathcal{N}$ により規格化された一様な重ね合わせ状態

$$\mathcal{N} \sum_{\boldsymbol{\delta}} |\boldsymbol{\delta}\rangle \equiv \mathcal{N} \sum_{\delta_1} \ldots \sum_{\delta_d} |\delta_1\rangle \ldots |\delta_d\rangle \tag{2.4.48}$$

を用意する。次に（位相）オラクルと呼ばれる関数 f のクエリに対応する量子操作 $U_{2\pi Sf}$ を

$$U_{2\pi Sf}\left|\boldsymbol{\delta}\right\rangle = \exp(2\pi i S f(\boldsymbol{x}+\boldsymbol{\delta}))\left|\boldsymbol{\delta}\right\rangle \tag{2.4.49}$$

と定義し、生成した状態に施す。（ここで S をある正のスケール因子とした。）仮定である式 (2.4.47) により近似を行うと量子状態は

$$U_{2\pi Sf}\mathcal{N}\sum_{\boldsymbol{\delta}}\left|\boldsymbol{\delta}\right\rangle \approx \exp(2\pi i S f(\boldsymbol{x}))\mathcal{N}\sum_{\boldsymbol{\delta}}\exp(2\pi i S\nabla f\cdot\boldsymbol{\delta})\left|\boldsymbol{\delta}\right\rangle \tag{2.4.50}$$

となる。そして d 個のベクトル成分それぞれに対し逆量子フーリエ変換を施すと、S と逆量子フーリエ変換のサイズにより定まるスケール因子 s を用いて

$$\exp(2\pi i S f(\boldsymbol{x}))\left|s\nabla f\right\rangle \tag{2.4.51}$$

という状態が得られ、最後に観測を行うと大域的位相因子 $\exp(2\pi i S f(\boldsymbol{x}))$ を除き $s\nabla f$ が得られる[*11]。

[*11] このようにあらかじめ定めたスケール因子 S から決まる因子 s と勾配 ∇f の積 $s\nabla f$ として勾配に関する情報が得られるため、実際には勾配を推定する空間領域において $s\nabla f$ がビットで正しく表現できる程度に勾配が十分小さいという仮定が必要である。

第 3 章

最小値探索量子アルゴリズムによる最適化問題への応用

ここでは最小値探索量子アルゴリズムの使用を前提にして、量子振幅増幅による古典機械学習モデルの学習を行う。

3.1 計算基底における四則演算を用いた古典的パーセプトロンモデル

初めに古典的なパーセプトロンモデルとして線形分離可能なデータに対する 2 クラス分類器

$$f\left(\boldsymbol{X} \equiv (-1, \boldsymbol{x})^{\intercal}; \boldsymbol{W} \equiv (w_0, \boldsymbol{w})\right) = \theta(\boldsymbol{W}\boldsymbol{X}) = \theta(\boldsymbol{w}^{\intercal} \cdot \boldsymbol{x} - w_0) \tag{3.1.1}$$

$$\theta(x) \equiv \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases} \tag{3.1.2}$$

を考える。ここで重みベクトル $\boldsymbol{w}$ と特徴量ベクトル $\boldsymbol{x}$ はあるビット数で 2 の補数表現されているとし、ベクトルの次元を n_1 とする。これは入力層におけるニューロンの数に一致する。しかしながら、内積 $\boldsymbol{w}^{\intercal} \cdot \boldsymbol{x}$ における乗算 $\times$ は 2 の補数表現を用いた可逆計算においてビット表現を考慮した際本質的な演算ではない。その代わりに符号反転を含めた乗算 $\times(-1)\times$ がビット表現に適した形式（付録 B.2.2 参照）であるから、上のパーセプトロンモデルを以下のように修正する。

$$f(\boldsymbol{X}; \boldsymbol{W}) = \bar{\theta}(\boldsymbol{W}\boldsymbol{X}) = \bar{\theta}(\boldsymbol{w}^{\intercal} \cdot \boldsymbol{x} - w_0) \tag{3.1.3}$$

$$\bar{\theta}\left(x\right) \equiv 1 \oplus \theta(-x) = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases} \tag{3.1.4}$$

ここで $\oplus$ は排他的論理和を意味する。すると、

$$f(\boldsymbol{X}; \boldsymbol{W}) = 1 \oplus \theta(w_0 - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x}) = (\,2\text{ の補数表現における } w_0 - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x} \text{ の符号ビットの値}) \tag{3.1.5}$$

と 2 の補数表現での符号ビットと直接的に対応づけることが可能になる。

次に、モデル（ここではパーセプトロン）の学習を行うためのコスト関数（損失関数）を定義する。機械学習においてはモデルに学習させたいデータセットに対し、コスト関数の最小化を行うことで与えられたデータを再現するようなモデルを構築することを学習と呼ぶ。自然数 N_{data} 個のデータを持つ学習用データセットを $\{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, N_{\mathrm{data}}\}$ とする。学習が適切に行われたかを評価するための尺度（コスト関数）として、精度 $Accuracy$ を

$$Accuracy \equiv \sum_{i=1}^{N_{\mathrm{data}}} |(y_i - f(\boldsymbol{X}_i; \boldsymbol{W})) \oplus 1| = \sum_{i=1}^{N_{\mathrm{data}}} |y_i \oplus f(\boldsymbol{X}_i; \boldsymbol{W}) \oplus 1| \tag{3.1.6}$$

と定義する。$Accuracy$ は 0 から N_{data} までの整数値を取る。学習によりもし学習用データセットを完全に正しく 2 値分類できれば $Accuracy$ は N_{data} と一致し、逆に全く分類できなければ 0 になる。ただし量子回路上では 2 の補数表現を基本として用いるため、$Accuracy$ のビット表現を可能な限り行うために $b_{Accuracy} \equiv \lceil \log(N_{\mathrm{data}}+1) \rceil$ ビットを用いて

$$Accuracy \leftarrow Accuracy - 2^{b_{Accuracy}-1} \tag{3.1.7}$$

と定数だけシフトして定義する。

パーセプトロンの学習のために、ある整数の閾値 $0 \leq A_{\mathrm{th}} \leq 2^{b_{Accuracy}} - 1$ を設定し $Accuracy$ がそれを上回るか否かの判定を行うブール値関数 $\chi\colon \mathbb{Z} \to \{0,1\}$

$$\chi(Accuracy, A_{\mathrm{th}}) \equiv \begin{cases} 1 & (Accuracy > A_{\mathrm{th}}) \\ 0 & (Accuracy \leqq A_{\mathrm{th}}) \end{cases} \tag{3.1.8}$$

を定義する。ただし $Accuracy$ と同じくビット数を抑えるため量子回路内では、$A_{\mathrm{th}} \leftarrow A_{\mathrm{th}} - 2^{b_{Accuracy}-1}$ と定数だけシフトさせて表現する。χ によって探索条件を設定し量子振幅増幅アルゴリズムを用いて学習を行う。これらの前提でパーセプトロンを量子回路内で計算基底を用いて構成する。ここでは以下の戦略で回路の構成を行う。

1. 必要な量子ビット数を最小化する。
2. 必要無くなればユニタリ逆演算をして対応するレジスタを初期化する。

これに基づき

1. 逐次的に $(w_0 - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x}_i)$ を計算する。
2. $(\boldsymbol{x}_i, y_i), A_{\mathrm{th}}, (w_0 - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x}_i)$ の入力と初期化を繰り返す。

ことを行う。以下必要なビット数をまとめる。

- $b_{\boldsymbol{x}}$：$\boldsymbol{x}$ の各成分のビット数
- b_{w_0}：w_0 のビット数
- $b_{\boldsymbol{w}}$：$\boldsymbol{w}$ の各成分のビット数
- $b_{Accuracy} = \lceil \log(N_{\mathrm{data}}+1) \rceil$：$Accuracy$ のビット数
- $b_{\mathrm{ancilla}} = \max(b_{\boldsymbol{x}}, b_{Accuracy}+1)$：学習用データ $(\boldsymbol{x}_i, y_i)$、閾値 A_{th} の入力またはその他のための作業用量子レジスタのビット数
- $b_{(w_0 - \boldsymbol{w}^{\intercal}\cdot\boldsymbol{x})} \equiv b_{\boldsymbol{x}} + b_{\boldsymbol{w}} - 1 + \lceil \log n_1 \rceil$：$(w_0 - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x})$ のとりうる数に対応したビット数

これに基づき量子レジスタを以下のように用意し $\mathrm{Ancilla}, \boldsymbol{w} = (w_1, \ldots, w_{n_1}), w_0, (w_0 - \boldsymbol{w}^{\intercal} \cdot \boldsymbol{x}), Accuracy$ と呼ぶ。

量子振幅増幅を用いた回路を構成する。**Algorithm 4** にその手続きをまとめた。増幅の判定のための $\boldsymbol{S}_0$ の構成は補題 2.4.4 の結果を利用している。ここで「符号ビットの値を加える」という表現を用いた際に、1 ビットの 2 の補数表現では $1 \equiv -1$ であるから実際には「符号ビットの値を引く」という操作をしている点に注意する。step:24 において Ancilla の第 0 ビットを飛ばして利用しているのは、引き算 QNModSub による $(0, a, b) \mapsto (a-b, b)$ と桁の変化を考慮しているためである。（付録 B 参照）

実際にパーセプトロン学習のためのシミュレーションを行うために、以下のようなデータ（図 3.2a 参照）と各ビット数を用意した。

- $n_1 = 2,\ \{\boldsymbol{x}_i = (x_{i,1}, x_{i,2}),\ y_i \in \{0,1\}\}_{i=1}^{N_{\mathrm{data}}}$
- $b_{\boldsymbol{x}} = 3,\ b_{\boldsymbol{w}} = 2,\ b_{w_0} = 3,\ N_{\mathrm{data}} = 7$

このデータを用い、閾値 $A_{\mathrm{th}} = 6 < 7 = N_{\mathrm{data}}$ として、増幅回数を変え量子振幅増幅を行った。以下に増幅された状

Algorithm 4 量子振幅増幅を用いたパーセプトロン学習のためのプログラム

Initialize:
  $\boldsymbol{\mathcal{A}}|0\rangle$ を用意する。
for $i = 0, \ldots, m$ **do**: ▷ Quantum Amplitude Amplification
  $\mathbf{Q} \equiv -\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}\mathbf{S}_\chi$ を系に作用させる。
end for
return 増幅（減衰）された $Accuracy > A_{\mathrm{th}}$ を満たす状態
procedure $\boldsymbol{\mathcal{A}}$ ▷ 解を増幅させるべきアルゴリズム
  アダマール変換をレジスタ $\boldsymbol{w}, w_0$ に行い、ありうるウェイトの状態を一様に重ね合わせる。
  レジスタ *Accuracy* の符号ビットを X ゲートにより反転。
  for $i = 1, \ldots, N_{\mathrm{data}}$ **do**:
    レジスタ $(w_0 - \boldsymbol{w}^\intercal \cdot \boldsymbol{x})$ に w_0 を加える。
    for $j = 1, \ldots, n_1$ **do**: ▷ calculate $(w_0 - \boldsymbol{w}^\intercal \cdot \boldsymbol{x})$
      $(\boldsymbol{x}_i)_j$ をレジスタ Ancilla に入力。
      $-(\boldsymbol{w})_j \times (\boldsymbol{x}_i)_j$ をレジスタ $(w_0 - \boldsymbol{w}^\intercal \cdot \boldsymbol{x})$ に入力。
      step 12 のユニタリ逆演算を行う。
    end for
    if $y_i = 0$ **then**
      レジスタ $(w_0 - \boldsymbol{w}^\intercal \cdot \boldsymbol{x})$ の符号ビットを X ゲートにより反転。
    end if
    レジスタ Accuracy に $(w_0 - \boldsymbol{w}^\intercal \cdot \boldsymbol{x})$ の符号ビットの値を加える。
    step 9 から step 18 までのユニタリ逆演算を行う。
  end for
end procedure
procedure $\boldsymbol{S}_\chi$ ▷ 解の判定
  A_{th} をレジスタ Ancilla の第 2 ビットから入力
  レジスタ Ancilla の第 1 ビットから以下 b_{Ancilla} 個の量子ビットにレジスタ *Accuracy* の値を差し引く。
  CZ ゲートをレジスタ Ancilla の第 1 ビットに施す。
  step 24 から step 25 までのユニタリ逆演算を行う。
end procedure
procedure $\boldsymbol{S}_0$
  X ゲートをレジスタ $\boldsymbol{w}, w_0$ の全ビットに施す。
  レジスタ $\boldsymbol{w}, w_0$ 全体のビット数に合うようにして多重制御 Z ゲートを施す。
  X ゲートをレジスタ $\boldsymbol{w}, w_0$ の全ビットに施す。
end procedure

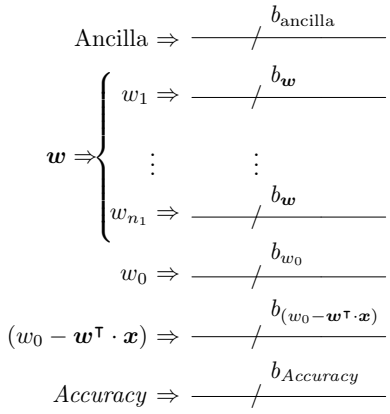
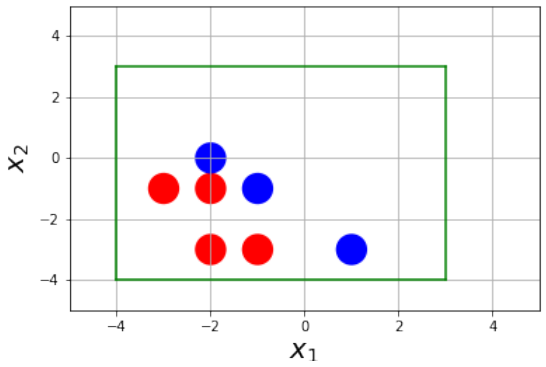
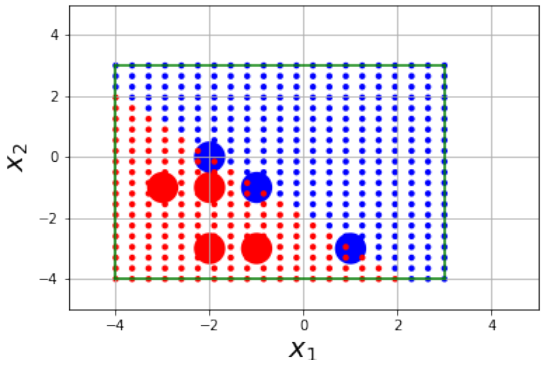


図 3.1: パーセプトロン学習のためのレジスタの配置



(a) 学習のためのデータセット



(b) 学習後のパーセプトロン

小さな赤、青マーカーは学習後のパーセプトロンにより分類されたデータを表す。

図 3.2: パーセプトロンによる 2 値分類

大きな赤、青マーカーはクラス 0、1 を表す。緑の枠線は $\boldsymbol{x}$ のビット数 $\boldsymbol{x}$ によって表現可能な $\boldsymbol{x}$ の範囲を表す。量子振幅増幅によって正しくデータを分類できるモデルが学習されていることが確認できる。

態の振幅を見るために Qiskit の *Statevector* シミュレーション[*1]を用いた結果を図 3.3 に示す。

量子振幅増幅によって、閾値 $A_{\text{th}} = 6$ を超える、特定のパラメータ $\boldsymbol{w} = \{-1, -1\}$, $w_0 = 2$ を持った状態のみが増幅されていることがわかる。得られたパラメータに対応するモデルの分類する領域を表現したものが図 3.2b に示されていおり、学習用データが分類できている。

[*1] 量子回路により得られる状態を数値計算し観測は伴わずに、理論的な出力分布を計算している

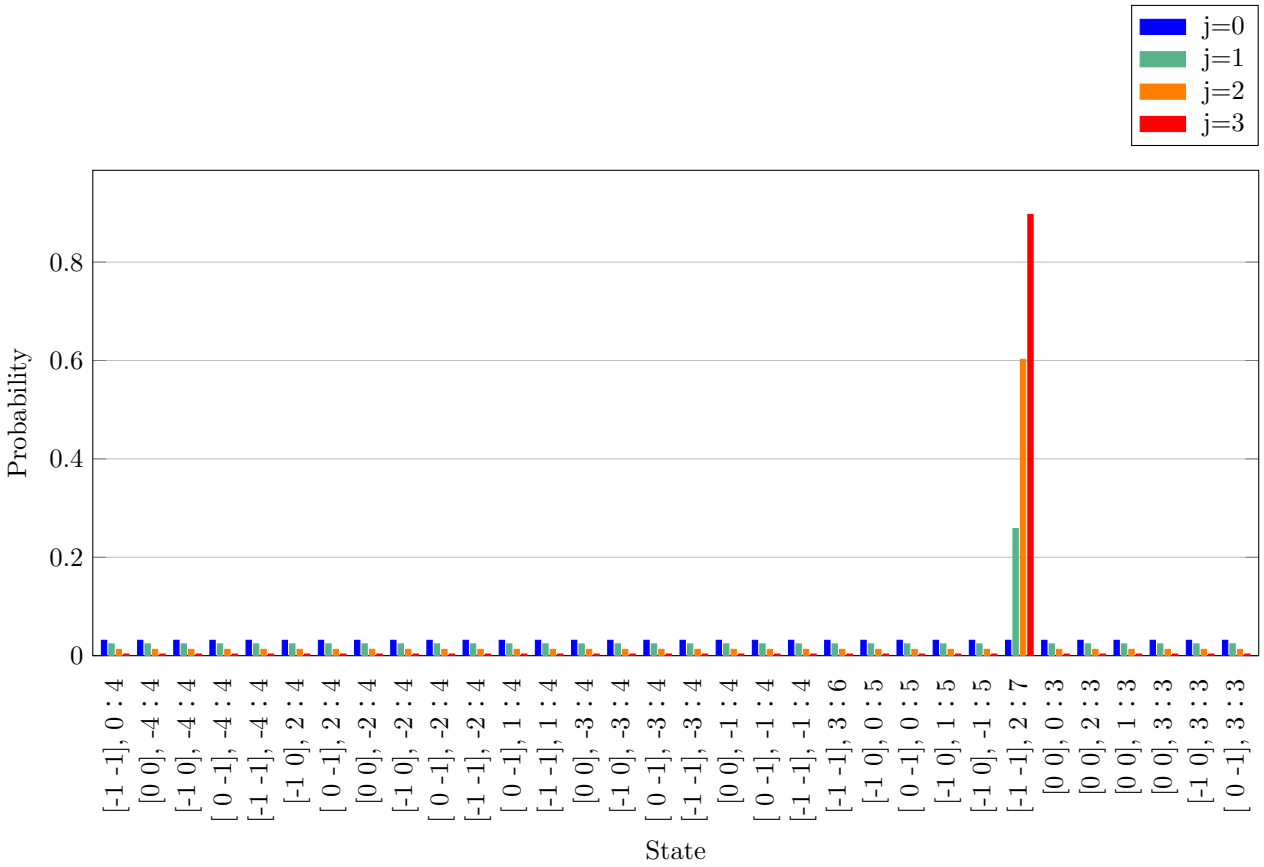


図 3.3: 量子振幅増幅による条件を満たすパラメーターの増幅

横軸はパラメーター $\boldsymbol{w}, w_0$ をもち、精度 $Accuracy$ を持った状態を表す。縦軸は観測される理論的な確率を表す。判例における j は増幅回数を意味する。

3.2　計算基底における四則演算を用いた古典的 FNN モデル

ここでは前節のパーセプトロンモデルを拡張し、隠れ層 1 つのフィードフォワード・ニューラルネットワーク (FNN、順伝搬型ニューラルネットワーク) モデルを構成し、量子回路内で実現する。ノーテーションは主に前節を踏襲する。

簡単のために以下活性化関数をステップ関数とする。そして出力は 1 つのみの 2 値分類のみ考える。入力層、隠れ層でのニューロン数を n_1, n_2 とおく。前節でも触れたが、ビット表現に適した演算としては、乗算ではなく符号反転

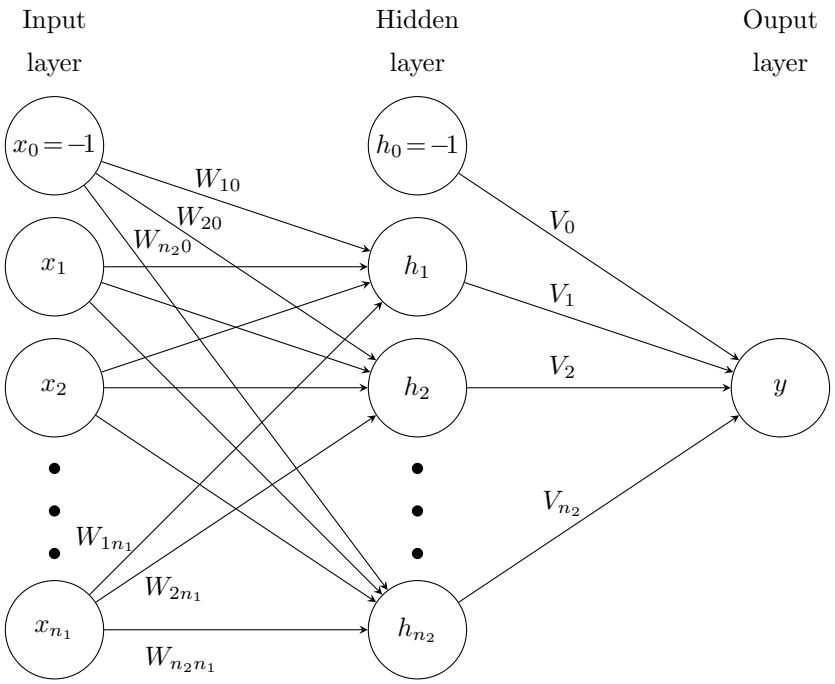


図 3.4: 2 値分類のための隠れ層 1 つ持つ FNN

を伴った乗算であるから、以下のようなものを考える。

$$\boldsymbol{X} \equiv (-1, \boldsymbol{x})^{\mathsf{T}} \tag{3.2.1}$$

$$\boldsymbol{V} \equiv (v_0, \ldots, v_{n_2}) \equiv (v_0, \boldsymbol{v}) \tag{3.2.2}$$

$$\boldsymbol{W} \equiv \begin{pmatrix} \boldsymbol{W}_1 \\ \vdots \\ \boldsymbol{W}_{n_2} \end{pmatrix} \tag{3.2.3}$$

$$\boldsymbol{W}_a \equiv (W_{a0}, W_{a1}, \ldots, W_{an_1}) \equiv (w_{a0}, \boldsymbol{w}_a) \ \ (a = 1, 2, \ldots, n_2) \tag{3.2.4}$$

$$\boldsymbol{H}(\boldsymbol{X}; \boldsymbol{W}) \equiv \begin{pmatrix} h_0 \\ \vdots \\ h_{n_2} \end{pmatrix} \equiv \begin{pmatrix} -1 \\ \boldsymbol{h} \end{pmatrix} \equiv \begin{pmatrix} -1 \\ \bar{\theta}(\boldsymbol{W}\boldsymbol{X}) \end{pmatrix}$$

$$f(\boldsymbol{X}; \boldsymbol{V}, \boldsymbol{W}) \equiv \theta(\boldsymbol{V}\boldsymbol{H}) \equiv \theta\left(\sum_{a=1}^{n_2} v_a \bar{\theta}\left(\sum_{i=1}^{n_1} w_{ai} x_i - w_{a0} \right) - v_0 \right) \tag{3.2.5}$$

この FNN に対応したグラフを図 3.4 に描いた。

$f(\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{W})$ がビット表現に沿った形になっていることを確認する。そのためにまず 1 を 1 ビットの 2 の補数表現で表現すると $1 \equiv -1$ であることに注意する。その事実を表現するために関数 $L_1 : \mathbb{B} \to \{0, -1\}$ を

$$L_1 = \begin{cases} 0 & (x = 0) \\ -1 & (x = 1) \end{cases} \tag{3.2.6}$$

とおく。すると、

$$\begin{aligned} f(\boldsymbol{x}; \boldsymbol{V}, \boldsymbol{W}) &= \theta\left(\sum_{a=1}^{n_2} v_a \left(w_{a0} - \sum_{i=0}^{n_1} w_{ai} x_i \text{の符号ビットの値} \right) - v_0 \right) \\ &= \left(\left\{ -\sum_{a=1}^{n_2} v_a L_1 \left(w_{a0} - \sum_{i=0}^{n_1} w_{ai} x_i \text{の符号ビットの値} \right) - v_0 \right\} \text{の符号ビットの値} \right) \oplus 1 \end{aligned} \tag{3.2.7}$$

と確かにビット（または 2 の補数）表現に基づいた形になっている。よって $\boldsymbol{h}_* = -\boldsymbol{h}$ とする。

前節のパーセプトロンと同様に、学習を行うための尺度（コスト関数）として、精度 $Accuracy$ を

$$Accuracy \equiv \sum_{i=1}^{N_{\text{data}}} |(y_i - f(\boldsymbol{X}_i; \boldsymbol{V}, \boldsymbol{W})) \oplus 1| = \sum_{i=1}^{N_{\text{data}}} |y_i \oplus f(\boldsymbol{X}_i; \boldsymbol{V}, \boldsymbol{W}) \oplus 1| \tag{3.2.8}$$

定め、表現ビット数を $b_{Accuracy} \equiv \lceil \log(N_{\text{data}} + 1) \rceil$ とおく。また閾値 $0 \leq A_{\text{th}} \leq N_{\text{data}}$ を定め、式 (3.1.8) を精度が閾値を超えるかどうかの判定として用いる。そして $Accuracy, A_{\text{th}}$ は量子回路内では、定数 $2^{b_{Accuracy}-1}$ を差し引いたものとして表現する。

回路の構成方法もパーセプトロンの場合と同じ戦略をとることにする。すると回路内で行う内容としては

1. $(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}_i), \left(v_a \bar{\theta}(\cdots) - v_0\right)$ を逐次計算する。
2. $(\boldsymbol{x}_i, y_i), A_{\text{th}}, (w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}_i), (-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)$ の入力と初期化を繰り返す。
3. $(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}_i)$ の計算結果の符号のみを作業用レジスタに保存し、$(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}_i)$ が表現されたレジスタはユニタリ逆演算によって初期化する。作業用レジスタに保存された符号を $(-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)$ の計算時に取り出す。

となる。パーセプトロンの場合と異なるのは、$(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}_i)$ の計算結果として必要な符号のみを保存し、$(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}_i)$ と $(-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)$ の計算に用いるレジスタを共有することで、必要な量子ビット数を減らす。以下必要なビット数をまとめる。

- $b_{\boldsymbol{x}}$：$\boldsymbol{x}$ の各成分のビット数
- $b_{\boldsymbol{h}} = 1$：$\boldsymbol{h}$ または $\boldsymbol{h}_*$ の各成分のビット数。つまり隠れ層の出力ビット数で今は 1 。
- b_{w_0}：w_{a0} それぞれのビット数
- $b_{\boldsymbol{w}}$：$\boldsymbol{w}_a$ それぞれの各成分のビット数
- b_{v_0}：v_0 のビット数
- $b_{\boldsymbol{v}}$：$\boldsymbol{v}$ の各成分のビット数
- $b_{Accuracy} = \lceil \log(N_{\text{data}} + 1) \rceil$：$Accuracy$ のビット数
- $b_{(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x})} \equiv b_{\boldsymbol{x}} + b_{\boldsymbol{w}} - 1 + \lceil \log n_1 \rceil$：$(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x})$ のとりうる数に対応したビット数
- $b_{(-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)} \equiv b_{\boldsymbol{h}} + b_{\boldsymbol{v}} - 1 + \lceil \log n_2 \rceil$：$(-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)$ のとりうる数に対応したビット数
- $b_{(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}), (-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)} \equiv \max\left(b_{(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x})}, b_{(-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)}\right)$：$(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}), (-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)$ のとりうる数に対応したビット数
- $b_{\text{ancilla}} = \max(b_{\boldsymbol{x}} + n_1 b_{\boldsymbol{h}}, b_{Accuracy} + 1)$：閾値 A_{th} の入力、学習用データ $(\boldsymbol{x}_i, y_i)$ の $\boldsymbol{x}$ の 1 成分のみの入力と隠れ層の出力 $\boldsymbol{h}_*$ の保持、またはその他のための作業用レジスタのビット数

その上で図 3.5 のようにレジスタを配置する。レジスタの名前を図 3.5 における左から右への矢印に紐づけられたテキストと対応させ、レジスタ $\boldsymbol{w}_a$ などと呼ぶことにする。学習のための回路の構成方法を **Algorithm 5** に記す。前節と同じく $\boldsymbol{S}_0$ の構成は補題 2.4.4 の結果を利用している。ここでも「符号ビットの値を加える」という表現を用いた際に、1 ビットの 2 の補数表現では $1 \equiv -1$ であるから実際には「符号ビットの値を引く」という操作をしている点に注意する。

線形分離不可な学習用データセットとして、2 次元ベクトルを特徴量としてもち排他的論理和 (XOR) と似た振る舞いで 2 値分類されたもの（図 3.6a）を考える。入力層、隠れ層のニューロン数をそれぞれ $n_1 = 2, n_2 = 2$ とする。各ビット数を $b_{\boldsymbol{x}} = 2, b_{\boldsymbol{h}} = 1, b_{w_0} = 2, b_{\boldsymbol{w}} = 2, b_{v_0} = 1, b_{\boldsymbol{v}} = 1, b_{Accuracy} = 3, b_{(w_0 - \boldsymbol{w}^\intercal \boldsymbol{x}), (-v_0 - \boldsymbol{v}^\intercal \boldsymbol{h}_*)} = 5, b_{\text{ancilla}} = 4$ とし、全量子ビット数を 27 とした。この時ありうる $2^{15} = 32768$ 通りのパラーメーターの組み合わせを調べ上げることになる。すると振幅増幅なしの場合はそれぞれのパラメーター配置が確率 $1/2^{15} = 3.05 \times 10^{-5}$ の等しい確率で観測が得られることになる。

Algorithm 5　量子振幅増幅を用いた隠れ層 1 つの FNN の学習プログラム

Initialize:
　$\boldsymbol{\mathcal{A}}|0\rangle$ を用意する。
for $i = 0, \ldots, m$ **do**: ▷ Quantum Amplitude Amplification
　$\mathbf{Q} \equiv -\boldsymbol{\mathcal{A}}\mathbf{S}_0\boldsymbol{\mathcal{A}}^{-1}\mathbf{S}_\chi$ を系に作用させる。
end for
return 増幅（減衰）された $Accuracy > A_{\mathrm{th}}$ を満たす状態
procedure $\boldsymbol{\mathcal{A}}$ ▷ 解を増幅させるべきアルゴリズム
　アダマール変換をレジスタ $\boldsymbol{W}, W_0, \boldsymbol{V}, V_0$ に行い、ありうるウェイトの状態を一様に重ね合わせる。
　レジスタ *Accuracy* の符号ビットを X ゲートにより反転。
　for $p = 1, \ldots, N_{\mathrm{data}}$ **do**:
　　for $a = 1, \ldots, n_2$ **do**:
　　　for $i = 1, \ldots, n_1$ **do**: ▷ calculate $(w_{a0} - \boldsymbol{w}_a^\intercal \cdot \boldsymbol{x})$
　　　　$(\boldsymbol{x}_i)_p$ をレジスタ Ancilla に入力。
　　　　$-(\boldsymbol{w}_a)_j \times (\boldsymbol{x}_i)_p$ をレジスタ $(w_0 - \boldsymbol{w}^\intercal\boldsymbol{x})$ に入力。
　　　　step 12 のユニタリ逆演算を行う。
　　　end for
　　　レジスタ $(w_0 - \boldsymbol{w}^\intercal\boldsymbol{x})$ に w_{a0} を加える。
　　　レジスタ $(\boldsymbol{h}_*)_a$ にレジスタ $(w_0 - \boldsymbol{w}^\intercal\boldsymbol{x})$ の符号ビットを加える。
　　　step 11 から step16 までのユニタリ逆演算を行う。
　　　for $a = 1, \ldots, n_2$ **do**: ▷ calculate $(-v_0 - \boldsymbol{v}^\intercal\boldsymbol{h}_*)$
　　　　レジスタ $(-v_0 - \boldsymbol{v}^\intercal\boldsymbol{h}_*)$ から $v_a(\boldsymbol{h}_*)_a$ を引く
　　　end for
　　　レジスタ $(-v_0 - \boldsymbol{v}^\intercal\boldsymbol{h}_*)$ から v_0 を引く
　　　if $y_i = 1$ **then**
　　　　レジスタ $(-v_0 - \boldsymbol{v}^\intercal\boldsymbol{h}_*)$ の符号ビットを X ゲートにより反転。
　　　end if
　　end for
　　レジスタ Accuracy にレジスタ $(-v_0 - \boldsymbol{v}^\intercal\boldsymbol{h}_*)$ の符号ビットの値を加える。
　　step 10 から step 26 までのユニタリ逆演算を行う。
　end for
end procedure
procedure $\boldsymbol{S}_\chi$ ▷ 解の判定
　A_{th} をレジスタ Ancilla の第 2 ビットから入力
　レジスタ Ancilla の第 1 ビットから以下 b_{Ancilla} 個の量子ビットにレジスタ *Accuracy* の値を差し引く。
　CZ ゲートをレジスタ Ancilla の第 1 ビットに施す。
　step 24 から step 25 までのユニタリ逆演算を行う。
end procedure
procedure $\boldsymbol{S}_0$
　X ゲートをレジスタ $\boldsymbol{w}_a, w_{a0}, \boldsymbol{v}, v_0$ の全ビットに施す。
　レジスタ $\boldsymbol{w}_a, w_{a0}, \boldsymbol{v}, v_0$ 全体のビット数に合うようにして多重制御 Z ゲートを施す。
　X ゲートをレジスタ $\boldsymbol{w}_a, w_{a0}, \boldsymbol{v}, v_0$ の全ビットに施す。
end procedure

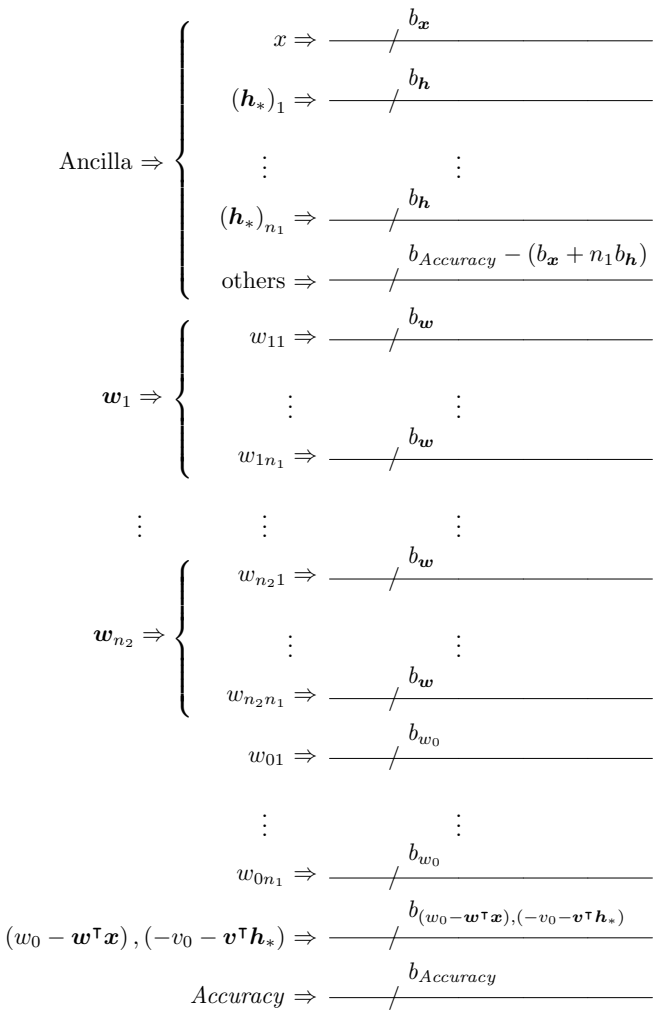
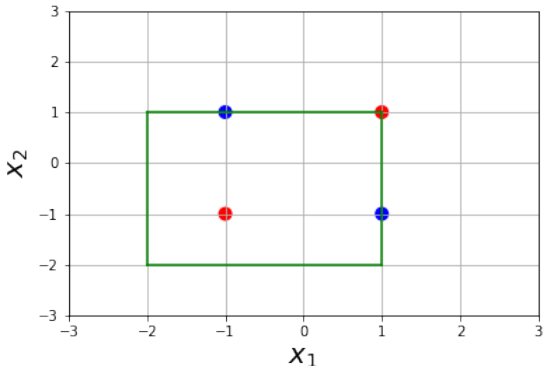
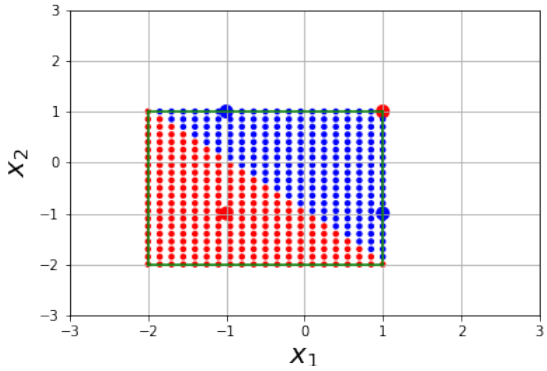


図 3.5: 隠れ層一つの FNN 学習のためのレジスタの配置

ここでは閾値 $A_{\mathrm{th}} = 3$ として $Accuracy = 4$ なる状態を増幅する。$Accuracy = 4$ なる状態は今の場合 42 通り存在し、増幅なしで ${}^{42}/_{2^{15}} = 1.28 \times 10^{-3} (\equiv \sin^2\theta$とおく) の確率で得られる。増幅回数 1 として振幅増幅を行うと、$Accuracy = 4$ なる状態が得られる確率は $\sin^2(2\cdot 1 + 1)\theta (\approx 9\sin^2\theta) = 1.15 \times 10^{-2}$ となる。増幅された状態のうちの 3 つを図 3.6b, 図 3.6c, 図 3.6d に示した。

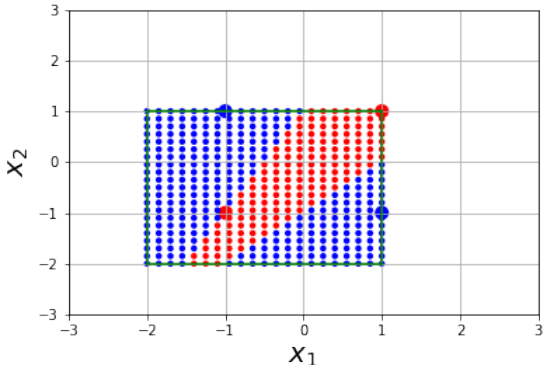


(a) 学習のためのデータセット



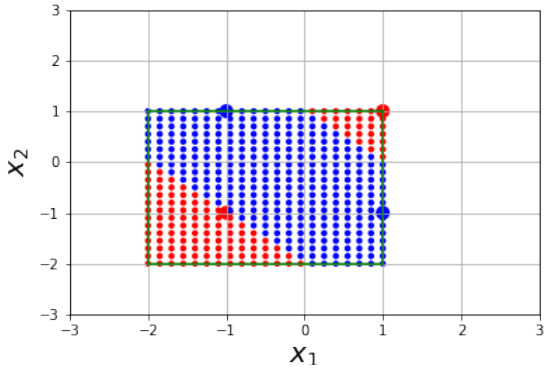
(b) 学習後の FNN の一つ

$\boldsymbol{w}_1=(-1,-1)\,,\boldsymbol{w}_2=(1,1)\,,w_0=(-2,-1)\,,$
$\boldsymbol{v}=(-1,-1)\,,v_0=-1$ に対応する



(c) 学習後の FNN の一つ

$\boldsymbol{w}_1=(-2,1)\,,\boldsymbol{w}_2=(1,-1)\,,w_0=(1,1)\,,$
$\boldsymbol{v}=(-1,-1)\,,v_0=0$ に対応する



(d) 学習後の FNN の一つ

$\boldsymbol{w}_1=(1,1)\,,\boldsymbol{w}_2=(-2,-2)\,,w_0=(-2,-2)\,,$
$\boldsymbol{v}=(-1,-1)\,,v_0=-1$ に対応する

図 3.6: FNN による 2 値分類

大きな赤、青マーカーはクラス 0、1 を表す。小さな赤、青マーカーは学習後の FNN により分類されたデータを表す。緑の枠線は $\boldsymbol{x}$ のビット数 $\boldsymbol{x}$ によって表現可能な $\boldsymbol{x}$ の範囲を表す。量子振幅増幅によって正しくデータを分類できるモデルが学習されていることが確認できる。

3.3 問題点

ここではこの手法の問題点について主に 3 つ説明する。

今のモデルでは量子ビットの計算基底を用いて情報を入力している。この方法の利点は、必要な古典計算を量子計算に置き換えることができさえすれば、基本的には古典計算に則った形式であるためモデルの直感的理解がしやすい点、エンコードを保ったまま量子重ね合わせ状態を生成したり他の量子計算と組み合わせる上でも非常に扱いやすい点、さらには一回の通常の観測自体によって情報を取り出すことのできる点などが挙げられる。

逆に欠点としては、他の情報のエンコード方法に比べ古典と同程度にビットが必要になってしまう点や、状態の準備にビット数とデータ数に関して線形の時間を要してしまう点である。2 つ目の欠点は、この手法では必要な量子ビットを抑えるために不要な情報はユニタリ逆変換を行いレジスタの初期化を行っているが、NN を多層にするにつ

れ、層数 L に関して古典計算の場合よりも指数的な増大が起きてしまう点である。L に関する計算量の指数的増大は古典計算でも起こるが、今のモデルではさらに 2^L の指数的増大が起きる。古典計算は非可逆であることが許され、情報の破棄が簡単に行えるが、Grover 探索は明らかに増幅する過程に可逆性が成り立つ必要があるからである。

最後の欠点は、古典的なパラメーター更新方法を考えればわかるが、必ずしもそのまま量子最小値探索をする必要がない。パーセプトロンを例に挙げる。学習段階 t ステップ目における重みパラメータを $\boldsymbol{w}(t)$ とした時、j 番目のデータ $(\boldsymbol{x}_j, y_j)$ に対する学習規則は

$$w_i(t+1) = w_i(t) + r\left(y_j - f_j(t)\right) x_{j,i} \tag{3.3.1}$$

$f_j(t)$ は t ステップ目でのモデル式、r は学習率と呼ばれる正の定数である。パーセプトロンの学習の収束性については、R を入力ベクトルの最大のノルム、データ間のギャップであるマージン γ に対し学習可能なデータセットならば、$O\left(R^2/\gamma^2\right)$ のステップで学習可能であることが 1960 年代から知られている [41]。そのため明らかに指数個のパラメーター空間から全探索する必要がない。それは誤差逆伝播法という学習規則を持つ FNN においても同じである。しかしだからといって量子探索アルゴリズムが不要なのではなく、文献 [42] のように古典計算によるパーセプトロン学習よりも早く収束するアルゴリズムが Grover 探索を用いて提案されている。よって最適な古典アルゴリズム内に量子探索を適用できるような部分を見出すことが重要なのである。これに関しては第 5 章へと議論が続く。

第 4 章

量子勾配推定アルゴリズムによる最適化問題への応用

4.1 整数係数・整数変数多項式に対する新しい量子勾配推定アルゴリズム

考案した勾配推定量子アルゴリズムについて紹介する前に、既存の勾配計算の手法について振り返る。基本的には勾配の定義である式 (2.4.46) から、勾配を計算する座標とそこから微小変位をとった座標での関数値 f を参照しそれらの差分（式 (2.4.47) 参照）を取ることで、近似的に勾配を得ている。この考え方は Jordan の手法で用いられており、それをさらに一般化した手法として中心差分

$$f(\boldsymbol{x}+\boldsymbol{\delta})-f(\boldsymbol{x}-\boldsymbol{\delta})=2\nabla f\cdot\boldsymbol{\delta}+O(\|\boldsymbol{\delta}\|^2) \tag{4.1.1}$$

やより高次の差分を用いることで一般の次数の多項式関数や滑らかな関数に対して効率的に勾配を計算することができる [17]。繰り返しになるがこれらは全て勾配以外の高次の項を微小量として近似していると言える。

そこで本研究では、関数がテイラー展開可能であると仮定して勾配以外の高次項を除く方法として、位相のもつ 2π の整数倍に関する周期性に着目した。具体的には関数値 f を位相に入力し、差分を取るための変位 δ に関して以下のような重ね合わせ状態を作ることで、テイラー展開時の係数が整数であれば自然数 m に対し

$$\begin{aligned}&\mathcal{N}\sum_{\boldsymbol{\delta}}\exp\left(2\pi i\ \frac{f(\boldsymbol{x}+2^m\boldsymbol{\delta})}{2^{2m}}\right)|\boldsymbol{\delta}\rangle\\&=\mathcal{N}\sum_{\boldsymbol{\delta}}\exp\left(2\pi i\ \left(\frac{f(\boldsymbol{x})}{2^{2m}}+\frac{\nabla f\cdot\boldsymbol{\delta}}{2^m}+O(\|\boldsymbol{\delta}\|^2)\right)\right)|\boldsymbol{\delta}\rangle\end{aligned} \tag{4.1.2}$$

において $\|\boldsymbol{\delta}\|^2$ の項が 2π の整数倍の位相として消失する。δ に依存しない上式の第一項については Jordan の手法と同じく大域的位相として無視され、d 個の逆量子フーリエ変換により勾配が得られる。この手法ではある座標における関数の形状を部分的に決定することで勾配を求めているので、関数形の決定問題と関連しているとも言える [43]。

以上の考察を主軸に考案した勾配推定量子アルゴリズムについて以下説明する。前提として数は 2 の補数表現により表現されているとする。仮定として d 変数のブラックボックスな整数変数・整数係数多項式関数 f が与えられていてかつ、各変数が m ビットで表現された整数値を取る関数 $f\colon \mathbb{Z}_{2^m}^d\to\mathbb{Z}$ を考える。所望の勾配を計算する座標を $\boldsymbol{x}=(x_1,\ldots,x_d)\in\mathbb{Z}_{2^m}^d$、記号の簡略化のため $n=\lceil\frac{m}{2}\rceil$ とする。

初期状態として

$$\left(|0\rangle^{\otimes m-n}|x_1 \bmod 2^n\rangle\right)\otimes\ldots\otimes\left(|0\rangle^{\otimes m-n}|x_d \bmod 2^n\rangle\right) \tag{4.1.3}$$

を用意する。$x_i \bmod 2^n$ は m ビット表現における整数 x_i のビット列の 2^0 の桁から 2^{n-1} の桁までのビット列のみを

取り出すことに対応する。次に部分的にアダマール変換 $\bigotimes_{j=1}^{d}(H^{\otimes m-n}\otimes I^{\otimes n})$ を行い、規格化因子 $\mathcal{N}$ を無視して状態

$$\begin{aligned}
&\left(\sum_{\delta_1}|\delta_1\rangle\,|x_1 \bmod 2^n\rangle\right)\otimes\ldots\otimes\left(\sum_{\delta_d}|\delta_d\rangle\,|x_d \bmod 2^n\rangle\right)\\
&\equiv\left(\sum_{\delta_1}|x_1+2^n\delta_1 \bmod 2^m\rangle\right)\otimes\ldots\otimes\left(\sum_{\delta_d}|x_d+2^n\delta_d \bmod 2^m\rangle\right)\\
&\equiv\sum_{\boldsymbol{\delta}}|\boldsymbol{x}+2^n\boldsymbol{\delta} \bmod 2^m\rangle
\end{aligned}\tag{4.1.4}$$

を作る。各途中式で $\delta_i\in\{-2^{m-n-1},\ldots,2^{m-n-1}-1\}$ とした。関数 f に対応するオラクルとして

$$O_{2\pi Sf}\,|\boldsymbol{x}\rangle=\exp(2\pi iSf(\boldsymbol{x}))\,|\boldsymbol{x}\rangle\tag{4.1.5}$$

を選択する。（オラクルの構成方法については付録 B.2.4 を参照されたい。）スケール因子 $S=\frac{1}{2^m}$ としたものをアダマール変換に続いて作用させた状態は、整数変数・整数係数多項式であるという仮定から大域的位相因子を除き、状態

$$\begin{aligned}
&\mathcal{N}\sum_{\boldsymbol{\delta}}\exp\left(2\pi i\frac{1}{2^{m-n}}\boldsymbol{\delta}\cdot\nabla f\right)|\boldsymbol{x}+2^n\boldsymbol{\delta} \bmod 2^m\rangle\\
&\equiv\mathcal{N}\left(\sum_{\delta_1}\exp\left(2\pi i\frac{1}{2^{m-n}}\delta_1(\nabla f)_1\right)|\delta_1\rangle\right)|x_1 \bmod 2^n\rangle\otimes\ldots\\
&\quad\otimes\left(\sum_{\delta_d}\exp\left(2\pi i\frac{1}{2^{m-n}}\delta_d(\nabla f)_d\right)|\delta_d\rangle\right)|x_d \bmod 2^n\rangle
\end{aligned}\tag{4.1.6}$$

と一致する。なぜなら、f が整数係数多項式で整数変数を持つので Taylor 展開時の多項式の係数は整数であり、また $\boldsymbol{x}_0'-\boldsymbol{x}_0\equiv\boldsymbol{0} \bmod 2^n$ とすると、

$$\begin{aligned}
f(\boldsymbol{x}_0'+2^n\boldsymbol{\delta})&=f(\boldsymbol{x}_0+(\boldsymbol{x}_0'-\boldsymbol{x}_0)+2^n\boldsymbol{\delta})\\
&=f(\boldsymbol{x}_0)+(\boldsymbol{x}_0'-\boldsymbol{x}_0+2^n\boldsymbol{\delta})\cdot\nabla f|_{\boldsymbol{x}=\boldsymbol{x}_0}+2^{2n}p(\boldsymbol{\delta},\boldsymbol{x}_0'-\boldsymbol{x}_0)\\
&=(\text{independent of }\boldsymbol{\delta})+2^n\boldsymbol{\delta}\cdot\nabla f|_{\boldsymbol{x}=\boldsymbol{x}_0}+2^{2n}p(\boldsymbol{\delta},\boldsymbol{x}_0'-\boldsymbol{x}_0)
\end{aligned}$$

ここで $p(\boldsymbol{\delta},\boldsymbol{x}_0'-\boldsymbol{x}_0)$ は $\boldsymbol{\delta},\boldsymbol{x}_0'-\boldsymbol{x}_0$ に関して二次以上で整数値をとる多項式とする。よって

$$\therefore f(\boldsymbol{x}_0'+2^n\boldsymbol{\delta})\equiv(\text{independent of }\boldsymbol{\delta})+2^n\boldsymbol{\delta}\cdot\nabla f|_{\boldsymbol{x}=\boldsymbol{x}_0}\ (\bmod 2^m)$$

であり、第一項は大域的位相として無視されるからである。

式 (4.1.6) の $\boldsymbol{\delta}\cdot\nabla f$ を含む位相部分の値は、十分なビット数（$m-n$ ビット以上）で 2 の補数表現された $\boldsymbol{\delta},\nabla f$ のビット列から定まる 2 進数表現における値を代入しても不変である。よって $\boldsymbol{\delta}\cdot\nabla f$ を 2 進数のビット列として扱うことで部分的に d 個のサイズ 2^{m-n} の逆量子フーリエ変換 $\bigotimes_{j=1}^{d}(\mathrm{IQFT}_{2^{m-n}}\otimes I^{\otimes n})$ を適用することが可能で状態

$$\left(|(\nabla f)_1\rangle\,|x_1 \bmod 2^n\rangle\right)\otimes\ldots\otimes\left(|(\nabla f)_d\rangle\,|x_d \bmod 2^n\rangle\right)\tag{4.1.7}$$

が得られる。最後に勾配のビット列に対応した量子ビットまたは量子レジスタを測定し、勾配 ∇f の各成分 $(\nabla f)_i$ が $m-n=\left\lfloor\frac{m}{2}\right\rfloor$ ビットで表現可能な数であれば、$\left\lfloor\frac{m}{2}\right\rfloor$ ビット表現で $(\nabla f)_i$ が正しく得ることができる。表現可能でなければ、十分なビット数における勾配の 2 の補数表現の 2^0 の桁から $2^{\left\lfloor\frac{m}{2}\right\rfloor-1}$ の桁までのビット列が得られる。また明らかにクエリ計算量は $O(1)$ であることが確認できる。観測を除いたこの手続きに対応する量子回路を図 4.1 に示した。

上述した勾配推定量子アルゴリズムの特徴について説明する。まず既存の手法との最も大きな差異は、関数の勾配以外の不必要な高次の項を位相の 2π の整数倍の等価性を利用して打ち消している点である。そしてこの手法に付随

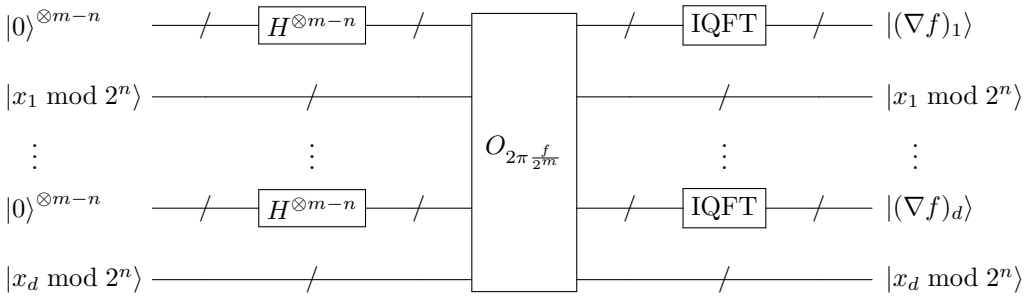


図 4.1: 勾配推定のための量子回路

し主に 2 つの利点と 2 つの欠点が存在する。一つ目の利点は、クエリ計算量が単に多項式関数の次元に依存しないだけでなく、多項式関数の次数にも依存しないという点であり、この手法特有の利点である。二つ目の利点としては、測定したときの分布が決定論的に一意に定まるという点である。このような一意な分布が得られる量子アルゴリズムは Deutsch-Jozsa[44] や Bernstein-Vazirani[45] の量子アルゴリズムといったものが有名だが、対照的に Grover の量子探索アルゴリズムといった観測分布が確率的であるものや、Grover 探索を応用し各ステップにおける測定結果に基づき動作する GAS アルゴリズム（**Algorithm 3** 参照）は、NISQ デバイスでは限定的にしか実行できない [46]。決定論的な出力分布を持つアルゴリズムは NISQ 時代においても部分的に検証可能性を与える。次節では実際に実機を用いて当該量子アルゴリズムを検証した結果を述べる。

一方一つ目の欠点は、強い仮定が要求される点である。関数が整数変数・整数係数多項式であるという要請は適用可能な対象を大きく狭めている。そして関数のブラックボックス性という仮定も実際に適用可能な問題を限定する。関数がブラックボックスであるという点は、関数の陽な表現や内部構造を知る必要がないという他のアルゴリズムを組み合わせる場合に便利でありつつも、今回特に適用可能な関数は多項式であるということから関数の陽な表現を知っていれば関数をあらかじめ解析的に微分したものを用いて勾配を計算すればよく、提案したアルゴリズムの有効性が明確ではなくなる。これに関してはのちの応用に関する第 4.3 節で議論を行う。二つ目の欠点としては、勾配 ∇f のビット表現が各成分ごとに、変数の表現ビット数の約半分でしか得られず、勾配の各成分の絶対値 $|(\nabla f)_i|$ が十分小さい時のみでしか、正しい結果を得られず符号でさえも適切か不明であるという点である。そして適用可能な関数の勾配の絶対値は大半の座標においてその条件を満たしていないと予想される。

4.2　シミュレーションと実験結果

考案したアルゴリズムを検証するため、以下の 3 つの関数 $f_1,\ f_2,\ f_3$、ビット数を決める m、変数の数 d と勾配を計算する座標 $\boldsymbol{x_0}$ を選択した。

$$
\begin{aligned}
&f_1(x) = -3x,\ x_0 = 1,\ m = 4,\ d = 1 \\
&\rightarrow \nabla f|_{x=x_0} = -3 \bmod 16 \qquad (4.2.1)\\
&f_2(x) = -4x^2,\ x_0 = 1,\ m = 4,\ d = 1 \\
&\rightarrow \nabla f|_{x=x_0} = -8 \bmod 16 \qquad (4.2.2)\\
&f_3(x,y) = 4xy,\ \boldsymbol{x}_0 = (0,\ 1),\ m = 3,\ d = 2 \\
&\rightarrow \nabla f|_{\boldsymbol{x}=\boldsymbol{x_0}} = (4,\ 0) \equiv (-4 \bmod 8,\ 0 \bmod 8) \qquad (4.2.3)
\end{aligned}
$$

簡単のために 1 次関数と 2 次関数、2 変数関数を選んでいる。前述した通り、勾配が表現ビット数に収まらなかった場合は、正しい結果が得られないことに注意する。3 つの関数 $f_1,\ f_2,\ f_3$ に対応した結果を以下実験 1、2、3 と呼称する。

シミュレーションにより検証するため Qiskit[47] と呼ばれるソフトウェア開発キットを用い、ノイズなしの場合のシミュレーションと実機におけるノイズモデルを利用したシミュレーションを行った。ノイズモデルは実デバイスで報告されたキャリブレーション情報をもとに生成されており以下のノイズを仮定している。

- 各量子ビットの各基底ゲートのゲートエラー率
- 各量子ビットの各基底ゲートのゲート長
- 各量子ビットの T_1, T_2 緩和時間定数
- 各量子ビットにおける読み出しエラー率

各実験において神奈川県川崎市に建造された *ibm_kawasaki* と呼ばれる実機 [6] を用いた。実行時のトランスパイル[*1]された量子回路に関する特徴は表 4.1 にまとめた。各実験時における試行（Shots）回数は 8192 回とした。また一般に逆量子フーリエ変換の 2 量子ビットゲートによる構成では、SWAP ゲートによる入力となる状態基底のビット列の反転が含まれている。NISQ 時代ではノイズ源となるのでそれを避けるため、ビット列の反転を行わないよう変更し観測した勾配情報に対応したビット列が反転しているものとする。

特徴	Depth	Size	# of CX
実験 1	39	75	17
実験 2	48	165	36
実験 3	45	232	73

表 4.1: トランスパイル後の量子回路

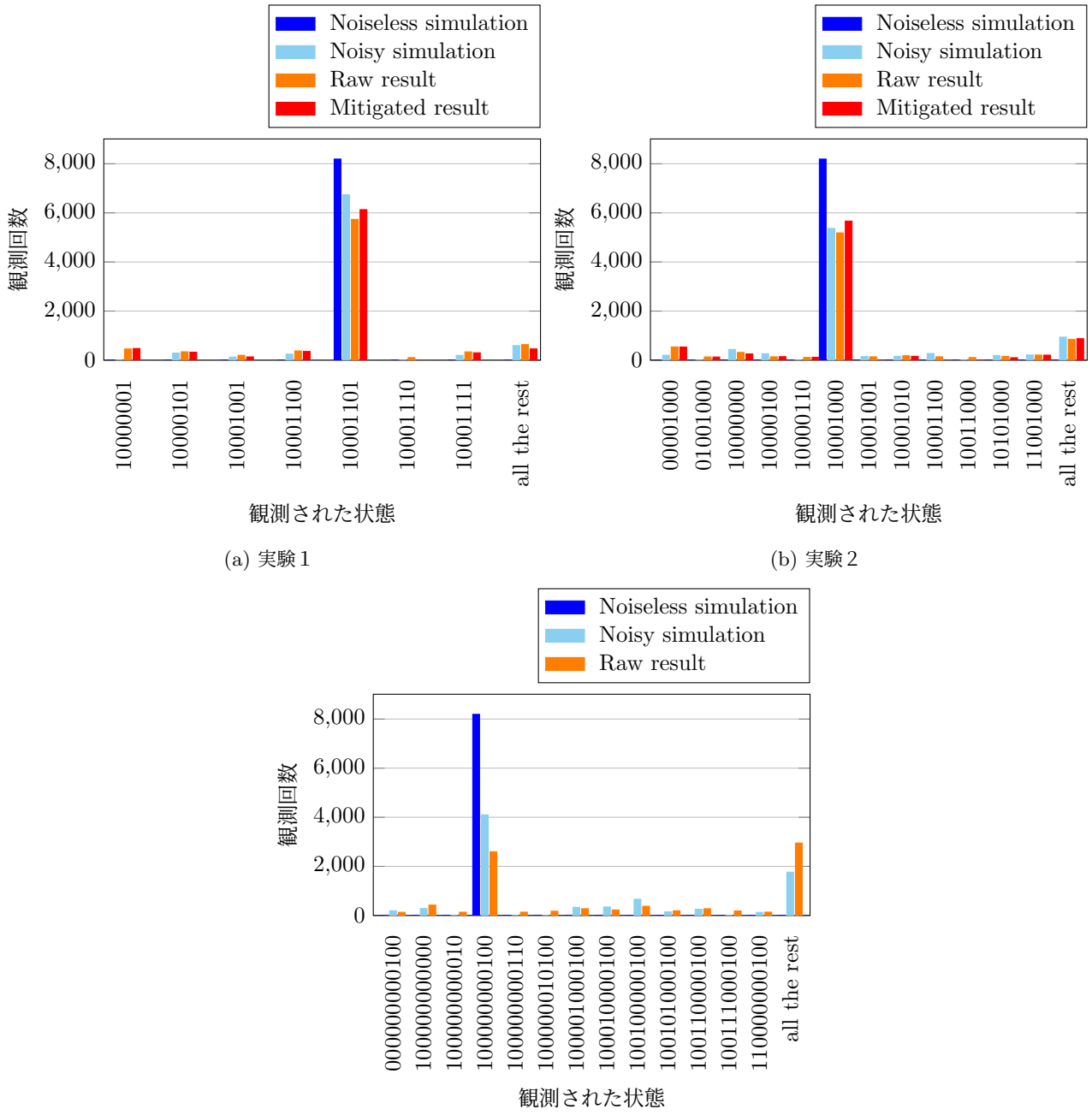
得られた実験結果に対し閾値 100 shots を設け、観測されたビット列を図 4.2 に示した。閾値に満たなかった状態は all the rest として表現している。ビット数の多い実験 3 を除いて、得られた結果に対し測定エラー逓減 [48] を行った。実験 1、2 においてはシミュレーションとほぼ近い結果が得られており 70% 以上の確率で正しい結果を得ることができた。実験 3 においてはアルゴリズムの成功率が 30% 程度に留まっているが、NISQ 時代としては比較的数の多い 12 量子ビットを用いても一定の結果が得られている。

そして図 4.3 においてノイズなしの理想的に観測されるビット列と観測結果とのハミング距離の分布を表現した。今の場合ハミング距離がノイズによるビット反転の数を指していて、それが二項分布に従うとすればハミング距離に関し近似的に指数的な減衰が起きていると解釈できる。各結果のハミング距離の平均値は表 4.2 にまとめた。量子ビットが比較的多い実験 3 においては比較的ビット反転が起きやすいことが見て取れる。

	Noisy simulation	Raw result	Mitigated Result
実験 1	0.240	0.450	0.373
実験 2	0.533	0.586	0.499
実験 3	0.802	1.437	—-

表 4.2: 平均ハミング距離

[*1] 量子回路を実行するには使用する量子コンピューターに適応させる必要があり、そのための操作を広くトランスパイル (transpile) という。具体的には、設計段階では量子回路において個々の量子ビット同士が接続（つまり 2 量子ビットゲートをどの 2 量子ビットの組み合わせに対しても作用できる）していると仮定しているが、現実には量子ビットの一部でしか接続されていない。そのため回路自体は変えずにその構成方法を変更する必要があり、付加的な SWAP ゲートが必要になりノイズの原因にもなっている。また実機上では用いることのゲートがあらかじめ定まっておりそれに合わせて記述する必要などがある。

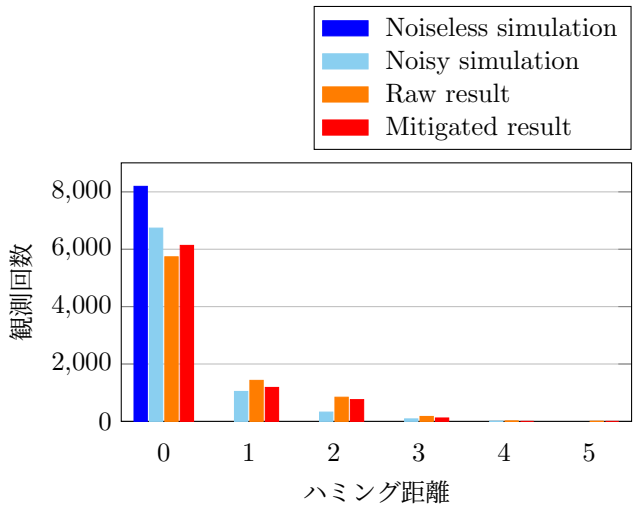


(a) 実験 1

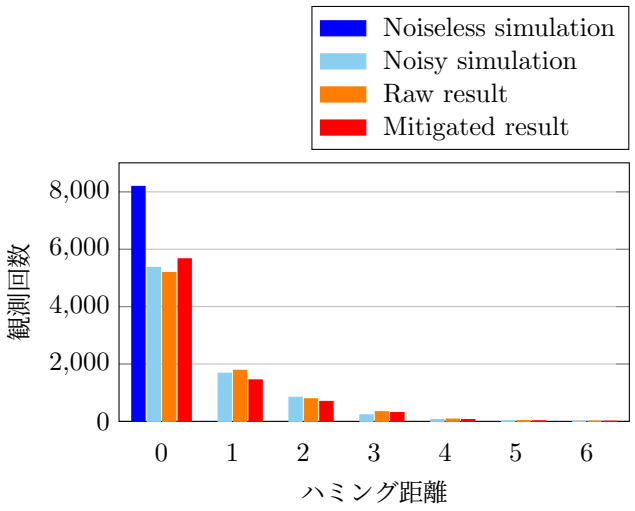
(b) 実験 2

(c) 実験 3

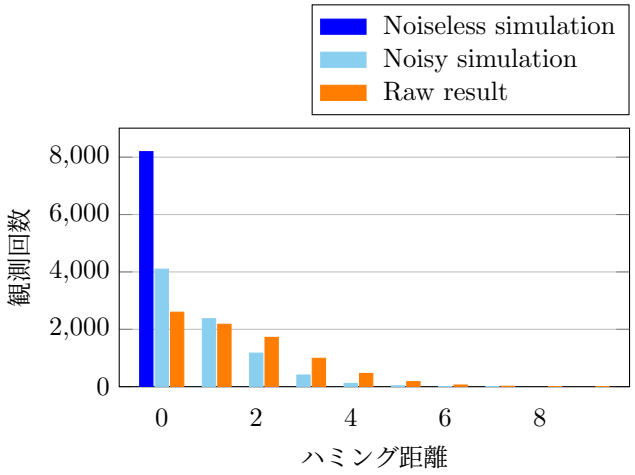
図 4.2: 観測されたビット列の分布



(a) 実験 1



(b) 実験 2



(c) 実験 3

図 4.3: 理想的な出力ビット列とのハミング距離の分布

4.3　勾配推定量子アルゴリズムの応用

ここでは考案したアルゴリズムを含め勾配推定量子アルゴリズムの問題点と応用先について議論する。

考案したアルゴリズムにおける整数変数の仮定はそれほど強い仮定ではない。その制限自体は単に勾配を計算する座標が整数という制限が与えるだけである。また勾配がビット表現に収まる程度に十分小さくなければならないという条件も、あらかじめ他の計算機などによりある程度関数の最適化を行い勾配が十分小さいという仮定をすれば良いので、それも致命的な問題とはならない。

次に多項式関数であることとブラックボックス性の仮定に関して議論するため、勾配のいくつかの計算方法について説明する。そもそも関数の勾配を計算する方法として主に 3 つの方法に分類することができる。

1. 数値微分・有限差分 (Numerical Differentiation・Finite Difference)
   微小区間 h で $\frac{f(x+h)-f(x)}{h}$ を計算する
2. 記号的微分 (Symbolic Differentiation)
   手計算もしくは、数式・記号処理により解析的に計算
3. 自動微分 (Automatic/algorithmic/computational Differentiation)
   主にプログラムで定義された複雑な関数をいくつかのブロックに分解し連鎖律を利用して導関数を求める

2 番目と 3 番目の手法については関数の表現が既知であることが前提となっている。そのような場合数値微分と同等もしくはより効率的に計算することができ、関数 f と同程度の計算量で勾配を計算することができる [49]。勾配推定量子アルゴリズムを使うにあたって重要なのは関数の表現は未知であり、関数の入力に対して出力のみしか知ることができないブラックボックスモデルを想定している点である。ブラックボックスモデルを考える利点としては関数の具体的な構造に左右されないという点がある。量子アルゴリズムの黎明期においては、個々の部分的なモジュールに依存せず複雑なアルゴリズムの計算量のみに調べるという目的において、クエリ計算量という計算量が定義された一つの動機であった。量子計算よりも先に発展していた計算機科学分野においては複雑なプログラムの内部構造にアクセスできないということが往々にして起こるのでブラックボックスの仮定がしばしば用いられる。

そうしたブラックボックス性が仮定できる最適化問題として、文献 [50] によると以下の４つの状況が考えられている。

1. 目的関数が、自動微分を適用できないようなコンピューターシミュレーションを通じて得られたものである
2. 何らかの実験・測定結果に基づいた目的関数であり、目的関数の数学的な表現が存在しない
3. 目的関数が何らかのノイズの影響を受けており、勾配に関する情報に対し信頼性が得られない
4. 仮に勾配情報が利用可能であったとしても、使い手がブラックボックス性を仮定する

４つ目の状況に対しては、効率的な最適化を目指す観点からして好ましくない用い方とされている。こうしたブラックボックス性を仮定できる実問題として同文献において３つの例が与えられている。

1. モンテカルロシミュレーションに基づいた結果を目的関数または制約条件が含み、試行ごとに異なる結果を与えうる場合
2. コンピューターシミュレーションのバグにより、目的関数が意味のある値を返さない可能性がある場合
3. 目的関数の出力が固定小数点数による丸め込み誤差の影響を受け、数値微分が意味をなさない場合

こうした４つの状況と３つの具体例を見ると、主にブラックボックス性が仮定できるのは、確率的不確かさやノイズ、その他未知の外的因子による影響を最適化問題が被る場合であると言える。

以上からブラックボックス性が仮定できそうな状況の例として広く “実験” が当てはまる可能性がある。より具体

的には、膨大な量の実験データに対し著名な HHL アルゴリズム [51] 等によるデータ解析を行う場合が該当する可能性がある。ただしフィッティングに関しては必ずしも勾配推定が必須ではない思われる。例えば最小二乗法による複雑でない関数形によるフィッティングについては、モデル式が与えられていて数値微分ではなくあらかじめ記号微分できるものが大半であるし、別途他の量子アルゴリズム [52] を用いるのが適切である。実験データにはノイズや確率的因子を含まれるが、解析という行為自体が「複雑なデータから有意な規則性を見出す」活動であり、つまりあらかじめ与えた幾つかの規則性の候補のいづれかに当てはめるということである。よって通常のフィッティングのような、与える規則性の候補自体が数式として書き下すことができてその構造が容易に解析できる場合、ブラックボックス性は成り立たない。逆に関数形が定かでない “フィッティング” に類するものとしては機械学習における学習が相当する。実際機械学習の分野では、学習におけるモデルの選択、ハイパーパラメータの調節においてはブラックボックス性が成立するため、“ブラックスボックス最適化” として研究が行われている。(広範な理解については文献 [53] 参照)

量子勾配推定を行って古典計算よりも優位性を得るには、ブラックボックス性だけでなく、さらに目的関数自体が数値微分が行えるような構造を持つ必要がある。ここまで提示した具体例では大抵ブラックボックス性と数値微分は相反するものとなっている。

しかし最適化問題で与えた目的関数を書き下すことができたとしても、その構造が容易には解析できず、同時に数値微分が可能なものは存在する。それが量子計算に現れるような量子操作や量子状態を含む目的関数である。勾配推定量子アルゴリズムに関する文献 [17] によると、主に量子シミュレーションやブラックボックスなユニタリーとしての量子オプティマイザーにより得られた目的関数に対し適用することが想定されており、３つの具体的な量子計算を含んだ問題への応用が提案されている。

1. 変分量子固有値ソルバー (Variational Quantum Eigensolver, VQE)[54]
2. 量子近似最適化アルゴリズム (Quantum Approximate Optimization Algorithm, QAOA)[55]
3. 量子自己符号化器 (Quantum Auto-Encoders, QAE)[56]

考案したアルゴリズムを適用するには、目的関数が量子計算に由来するものであった上でさらに整数係数多項式関数でなければならない。よって整数係数多項式関数によって、どの程度関数が近似できるかという問いも実用可能性を知るに当たって重要である。それに関して文献 [57] によって、次のような問題の定式化がなされ、それに続く２つの定理が示されている。

定義 4.3.1 (整数係数多項式関数による関数の近似問題)**.** 実数値関数 f が与えられていたとする。
任意の正の ϵ に対し、区間 I 上における一様ノルム $\|f-q\| < \epsilon$ を満たす有理整数[*2]を係数とした多項式関数 q が存在すれば、整数係数多項式関数により f は近似可能である。

定理 4.3.1 (区間の長さと近似の自明性)**.** 区間 I の長さが４以上であれば、区間 I 上での整数係数多項式関数による関数 f の近似は自明である、すなわち f 自身が整数係数多項式関数である。

有理整数係数のモニック多項式[*3]の複素根を代数的整数と呼び、与えられた代数的整数 α に対し α の最小多項式[*4]の根を α の共役と呼ぶ。

[*2] 広く一般に想像される整数であり、のちに現れる代数的整数と区別するために用いる。
[*3] 最高次係数が１である１変数多項式
[*4] 有理整数係数のモニック多項式で α を根としてもつ一意な最小の次数のもの

定理 4.3.2 (区間の長さと外挿による近似)**.** 厳密に 4 未満の長さを持つ区間 I 上の連続関数 f を考える。
関数 f が整数係数多項式によって近似可能であることは、代数的核 $J(I)$[*5]上で外挿された多項式関数の係数が全て整数であることと同値である。

考案した量子アルゴリズムは長さ 4 以上のような広い区間の上で多項式関数であることを必要としており、定理 4.3.1 から近似の自明性が成り立つ。よって任意の精度の近似はほぼ不可能であることがわかる。ある有限の誤差を許した近似の可能性については不明であるが、定理 4.3.1 で言う自明な近似しか行えないと予想される。

以上の考察から、考案した多項式関数に対する勾配推定量子アルゴリズムは、実問題において古典計算よりも有利となる状況が非常に考えにくい。なぜなら 3 つの量子計算を含んだ最適化問題における目的関数として整数係数の多項式関数が現れるとは考えにくいからである。そのため考案したアルゴリズムを実用の意味で発展させていくというよりは、本質的に重要な点である位相の対称性を利用するといった、提案したエンコーディング方法を進展させる方向性が主要になると思われる。近年注目されているエンコーディング方法としてはブロックエンコーディング [58] と呼ばれる手法が存在し、そうした状態の量子状態への入力や量子状態の変換を関数とどう結びつけるかが今後重要な研究対象になると思われる。

[*5] 代数的核 $J(I)$ を I に含まれる共役な代数的整数の全ての完全集合の和集合とする。例えば $J([-1,1]) = \{-1,0,1\}$, $J\left(\left[-\sqrt{2},\sqrt{2}\right]\right) = \{0,\pm1,\pm2\}$ である。

第 5 章

局所最適化が可能な最適化問題への量子計算の応用

第 3 章で述べたように、古典的機械学習において誤差逆伝播といった効率的な学習則が存在するため、そもそも可能なパラメータの組み合わせについて全探索する必要がない。そのため量子探索は、こうした問題の構造を利用した古典計算と同程度に効率的なアルゴリズムを使用した上でサンプリング部分に適用すべきである。実際そのように探索を含む多くの問題において量子探索が使用されている。だが同時に、そうした構造を持った問題に対し一般に成り立つ性質をあらかじめ知っておくことは、個別の問題に対する見通しをよくする上で重要である。また、現実的には観測結果に応じ、発見法的に最適化手法自体を更新することが考えられる。以下では具体的にどのような発見法的な改善をするかは述べないが、その点を考慮してアルゴリズムを構築する。さらに、実用上量子勾配推定は単独で用いることは少ないと考えられる。例えば、最適化の途中で停留点に到達し更新が停止した際に、古典計算では変数を再度初期化することを行う。これは multi-start 法（**Algorithm 9** 参照）といい、それに関し量子探索を適用した場合について議論する。

これらの議論と問題点から、何らかの局所的な構造を持った問題に対しサンプリングによる量子加速はどの程度得られるかということを、一般的に取り扱う枠組みが必要である。しかしこの問いかけに対し定量的かつ一般論として議論された例は少ない。本章では、量子的 multi-start 法として定式化を行いそれに伴う議論や問題点を浮き彫りにする。

具体的には第 2.4.1 節の量子振幅増幅による確率的アルゴリズムの、一部の発見的手法に関する議論を拡張し、第 2.4.2 節の最小値探索手法を基礎とした、一般の局所最適化を含んだ関数最小化アルゴリズム (**Algorithm 7**) を構成する。

それにあたって第 5.1 節では、その限定的なものまたは前段階として決定論的な局所探索の導入と局所解に停留した場合更新を停止させる場合のみ考え、具体例を提示し、議論を行う。その上で第 5.2 節ではより一般的な確率的局所最適化を含む最適化問題について議論する。その枠組みにおいて量子勾配推定を用いた時のさらなる量子加速について考察する。以下 $f\colon S \to \mathbb{R}$ を最小化の対象となる実数値関数とする。

5.1 量子振幅増幅と決定論的局所探索手法を用いた最適化

先行研究として古典的 multistart method（**Algorithm 9** 参照）が適用可能な問題に対し、文献 [22, 23] で量子最小値探索を適用することが考えられており、以下の **Algorithm 6** が考案されている。ここではそこでの議論を踏襲しつつもさらに議論を前進させる。

本章で述べたとおり以下決定論的な局所探索の場合で議論する。探索に確率的な振る舞いは存在せず、決定論的な更新をなすとする。また探索の “履歴” もしくは経路に依存せず探索を行うとする。さらに局所的最小値に陥った場

合更新は行われないとする。そうした探索として代表的なものが、山登り法である。

以上の前提の上で局所探索を関数 $b\colon S \to S$ として定式化する。局所探索を連続して L 回行う操作を $b\circ b\circ\cdots\circ b = b^L$ と表記する。局所探索 b に対し $i \in \{0,1,2,\ldots\}$ 番目に小さい局所的最小値 f_i を持った “凹地” を

$$S_i = \left\{x \in S \mid \forall L \in \mathbb{Z}_{\geq 0},\ f\left(b^L(x)\right) \geq f_i\right\} \cap \left\{x \in S \mid \exists L \in \mathbb{Z}_{\geq 0} \text{ such that } f\left(b^L(x)\right) = f_i\right\} \tag{5.1.1}$$

として定義する。（図 5.1 参照）

これは S_i 内の全ての x は局所探索 b により、局所的最小値 f_i へと陥ることを意味している。（この定義において局所的最小値 f_i が複数存在していたとしても、S_i を 1 つの凹地と呼んでいる。）同時に $\forall x \in S_i,\ f\left(b^L(x)\right) = f_i$ なる最小の L を L_i とする。つまり L_i は、i 番目の凹地 S_i にサンプリングされた点から探索を始め、局所的最小値 f_i をとるために必要な局所探索の最小の反復回数である。この時凹地のうち大域的最小値を含むものは S_0 であって

$$f_0 = f_{\min} = \min\{f(x) \mid x \in S\} \tag{5.1.2}$$

となっている。全ての領域内の点 $x \in S$ から局所的最小値に陥るまでに必要な局所探索の回数を

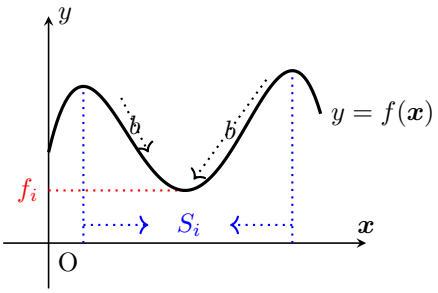


図 5.1: 局所探索 b に対する凹地 S_i の直感的イメージ

$$L_{\max} = \max\{L_0, L_1, L_2, \ldots\} \tag{5.1.3}$$

とする。これにより反復回数 L の局所探索を行った後の関数値の変化を直感的に図 5.2 に示す。これは十分な回数の局所的更新を繰り返すことで、S_i 内の全ての解の候補が関数値 f_i をとる局所解へ更新されることを意味する。

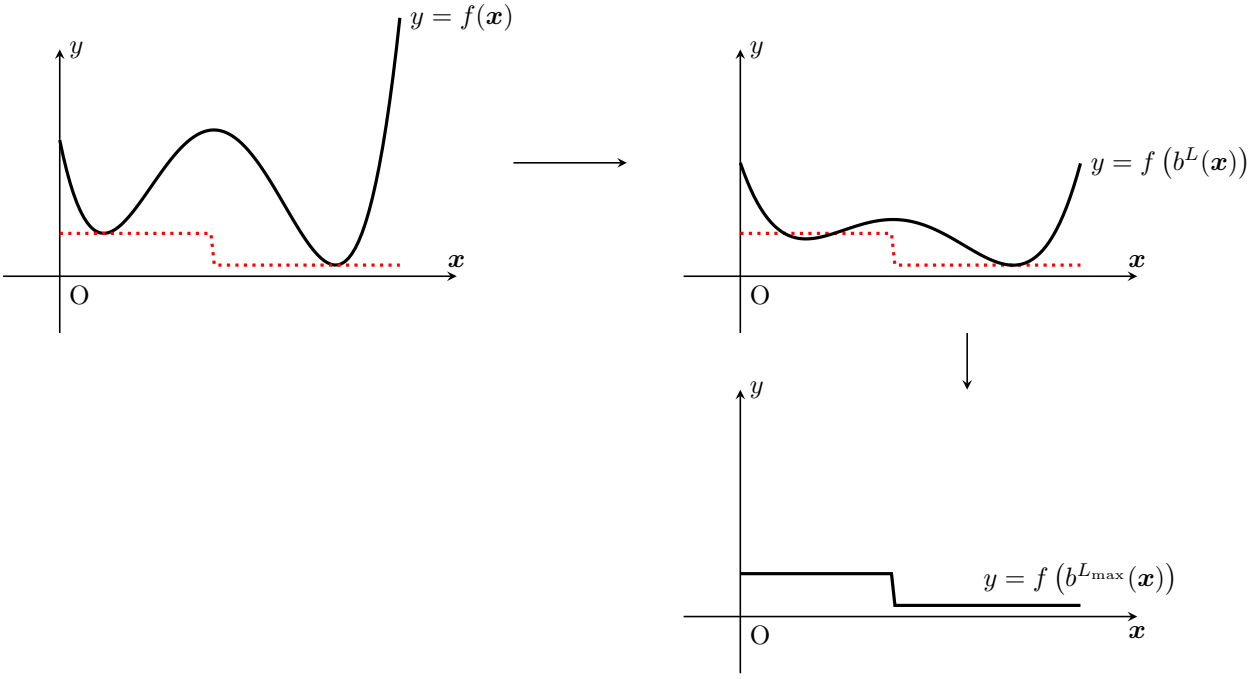


図 5.2: 局所探索 b の反復により更新された解に対する f の直感的イメージ

そして以下のように各集合を定義する。

$$S_{\min} = \{x \in S \mid f(x) = f_{\min}\} \tag{5.1.4}$$

$$(b^L)^{-1}(f_{\min}) = \left\{x \in S \mid f\left(b^L(x)\right) = f_{\min}\right\} \tag{5.1.5}$$

これらは順番に大域的最小値に対応する S の要素の集合、L 回の更新で大域的最小値を与えるような S の要素の集合である。そしてクエリにより計算量を評価するため、一回の局所探索のための古典・量子クエリ計算量 $C_{\mathrm{c}}(b,f),\ C_{\mathrm{q}}(b,f)$ とする。

以上の設定で multi-start 法によるクエリ量の評価を行う。古典的にはランダムなサンプリング $x \in S$ に対し反復回数 L の局所探索 b^L を行い、関数値 $f\left(b^L(x)\right)$ を得て、各サンプリングで得られた最小の値を最小値とする。そのため古典的な multi-start 法では、一回のサンプリングで更新後の解の候補に対する 1 回分のクエリも含め $1+LC_{\mathrm{c}}(b,f)$ のクエリを要し、最小値の獲得には局所探索後の最小点の数 $\left|(b^L)^{-1}(f_{\min})\right|$ に反比例したクエリが必要となる。

一方 Grover（量子振幅）増幅により改良された multi-start 法として、ランダムなサンプリング $x \in S$ に対し反復回数 L の局所最適化 b^L を行い、関数値 $f\left(b^L(x)\right)$ に関する最小値探索アルゴリズムを用いることを考える。この場合も探索後の解の数は $\left|(b^L)^{-1}(f_{\min})\right|$ に反比例するが、全体として 2 次の加速が起きるのでクエリ計算量は $\sqrt{\frac{|S|}{|(b^L)^{-1}(f_{\min})|}}$ に比例する。（手続きは **Algorithm 6** 参照[*1]）。しかし局所探索に関しては、増幅過程のユニタリ逆変換を考慮して反復回数と探索コストに線形なオーダーである最大 $1+2LC_{\mathrm{q}}(b,f)$[*2]のクエリが必要になる。

Algorithm 6 Quantum Basin Hopper, **QBH**(λ, M, L)

$x \in S$ をランダムに選択する
x を局所的最適解 x_{cand} へ更新する
$y_{\mathrm{cand}} \leftarrow f(x)$
while $y_{\mathrm{cand}} \geq Y$ **do**
  $x \leftarrow x_{\mathrm{cand}},\ y \leftarrow y_{\mathrm{cand}},\ m = 1$
  while $y_{\mathrm{cand}} < Y$ or $m > M$ **do**
    $\{0, \ldots, \lceil m-1 \rceil\}$ から一様ランダムに整数 r を選択する
    GBS(Y, r, L) の出力を測定し、$x_{\mathrm{cand}}, y_{\mathrm{cand}}$ とする。
    $m \leftarrow \lambda m$ とする。
  end while
end while
procedure Grover Basin Search, **GBS**(Y, r, L)
  一様な重ね合わせ状態を生成する部分的なアダマール変換、目的関数 f のクエリ、局所探索また局所最適化による解候補の更新、閾値条件 $\chi(x;Y)$[*3]に関する状態の位相反転それぞれに対応するユニタリ演算を $U_H, U_f, U_b, U_{\chi(x;Y)}$ とする。
  回転数 r としてユニタリ演算 $U_{\chi(x;Y)} \circ U_f \circ U_b^L \circ U_H$ の Grover 増幅を行う。
end procedure

以上の考察の元に量子・古典における multi-start 法を用いた局所探索によって最小値が得られるまでのクエリ計算量を表 5.1 にまとめた。また最小値のサンプル点が得られる平均的サンプリング確率のオーダー評価を示した。最小値を含む凹地 S_0 にサンプリングされる確率は $\Theta\left(\sqrt{\frac{|(b^L)^{-1}(f_{\min})|}{|S|}}\right)$ であることがわかる。つまり、局所探索によって最小値を含む凹地 S_0 へのサンプリング確率を増大させる効果があると言える。

ここでは古典的な探索のクエリコスト $C_{\mathrm{c}}(b,f)$ と量子的な探索のクエリコスト $C_{\mathrm{q}}(b,f)$ として分類したが、一般的な探索としては $C_{\mathrm{c}}(b,f) > C_{\mathrm{q}}(b,f)$ を実現するのは難しいと思われる。もし局所探索において $C_{\mathrm{c}}(b,f)$ 回のクエリによって解の数が既知な状態で探索するのであれば、量子振幅増幅によって $\Theta\left(\sqrt{C_{\mathrm{c}}(b,f)}\right)$ 回のクエリかつ 100% の確率で発見することが可能であるが、必ずしも局所探索において解の数が一定数とは限らないので、そうした解を

[*1] 凹地を飛び越える効果は量子最小値探索が担っているため、basin hopper と命名されている）
[*2] オーダー評価として本質的には瑣末なものであるが、説明上逆演算の存在を 2 倍の因子により明示している。
[*3] $\chi(x;Y) = 1\ (\text{if } x < Y), 0\ (\text{otherwise})$

L	古典的 multi-start 法		Grover（量子振幅）増幅による multi-start 法	
	最小化に必要なクエリ	サンプリング確率	最小化に必要なクエリ	サンプリング確率
0	$O\left(\frac{\lvert S\rvert}{\lvert S_{\min}\rvert}\right)$	$\frac{\lvert S_{\min}\rvert}{\lvert S\rvert}$	$O\left(\sqrt{\frac{\lvert S\rvert}{\lvert S_{\min}\rvert}}\right)$	$\Theta\left(\sqrt{\frac{\lvert S_{\min}\rvert}{\lvert S\rvert}}\right)$
$L(<L_0)$	$O\left((1+LC_{\mathrm{c}})\frac{\lvert S\rvert}{\lvert(b^L)^{-1}(f_{\min})\rvert}\right)$	$\frac{\lvert(b^L)^{-1}(f_{\min})\rvert}{\lvert S\rvert}$	$O\left((1+2LC_{\mathrm{q}})\sqrt{\frac{\lvert S\rvert}{\lvert(b^L)^{-1}(f_{\min})\rvert}}\right)$	$\Theta\left(\sqrt{\frac{\lvert(b^L)^{-1}(f_{\min})\rvert}{\lvert S\rvert}}\right)$
$L(\geq L_0)$	$O\left((1+LC_{\mathrm{c}})\frac{\lvert S\rvert}{\lvert S_0\rvert}\right)$	$\frac{\lvert S_0\rvert}{\lvert S\rvert}$	$O\left((1+2LC_{\mathrm{q}})\sqrt{\frac{\lvert S\rvert}{\lvert S_0\rvert}}\right)$	$\Theta\left(\sqrt{\frac{\lvert S_0\rvert}{\lvert S\rvert}}\right)$

表 5.1: multi-start 法により最小値を得るまでの局所探索回数に対する依存性

サンプリング確率とは大域的最小値がサンプリングされる確率を指す。ただし量子振幅増幅における場合は、閾値の更新とともにサンプリング確率は増大していきクエリ数にも依存するため、表の値はその平均的な値でかつオーダーでの評価となっている。

判定するオラクルを構成することは難しい。さらに仮に可能だとしても、局所探索の一回の更新において二次の加速しか生じず、加速が困難と考えられている反復回数 L に関する線形な依存性が問題になる。

また実際問題として重要なのは、局所探索の回数は何回が適切なのかという問題である。なぜなら古典的 multi-start 法においては局所探索ごとにその更新結果を観測しているため、解が更新されなければ新たにサンプリング点を取ればいいが、**Algorithm 6** ではあらかじめ定めた反復回数 L の局所探索を行う必要があり、一回の反復ごとに更新されているかの確認を観測によって確かめることができない。そのため、量子振幅増幅を用いた multi-start 法において、あらかじめ定めた L に対し

$$(1+2LC_{\mathrm{q}}(b,f))\sqrt{\frac{|S|}{|(b^L)^{-1}(f_{\min})|}}=O\left(\sqrt{\frac{|S|}{|S_{\min}|}}\right) \tag{5.1.6}$$

であることが、実際に効率的に反復回数 L の量子的探索を行う条件となる。その条件に関し問題点が 1 つ存在する。

それは一般に反復による最適化手法においては、そうした必要十分な反復回数を得るには問題の解を得るのと同程度に困難である場合が多い。少なくとも

$$\frac{|S|}{|S_0|}\leq\frac{|S|}{|(b^L)^{-1}(f_{\min})|}\leq\frac{|S|}{|S_{\min}|} \tag{5.1.7}$$

は成立するが、比

$$\frac{|S_{\min}|}{|(b^L)^{-1}(f_{\min})|} \tag{5.1.8}$$

の見積もりが難しく、仮にできたとしても、$|S_0|=\left|(b^L)^{-1}(f_{\min})\right|$ なる L_0 を事前に知ることは何らかの仮定なしには不可能であるからである。

この問題点に起因して探索回数が不十分である可能性がある。そうした場合には、観測ごとに得られた解の候補に対して、古典的に局所探索を局所解に陥るまで反復することで対応できる。

一方、L 回の局所最適化の更新の途中で更新が止まった場合には、残りのクエリ分は無駄になってしまう。更新が停止した場合の回避策としては、新たなサンプリング点を取り局所最適化の更新を行うことを L 回のクエリに到達するまで繰り返すことである。そして L 回の更新終了時に最も最小であったものを最小解の候補とするものである。しかしそれは、量子的な multi-start 法の一部に古典的な multi-start 法を用いるということになり、古典的な multi-start 法のサンプリングが増えるほど、量子の優位性は落ちていく。そのため必要十分な更新回数 L の見積もりがやはり重要である。

そうして反復回数を適切に設定することは困難であったが、$\frac{|S_{\min}|}{|(b^L)^{-1}(f_{\min})|}$ のスケーリングについて知るために、今の前提における具体例として d 次元の格子状のグリッド上で山登り法を考え、隣接する $2d$ 個の関数値を比較し更新し、更新されなければそのまま停止するような探索を採用し (つまり $C_{\mathrm{q}}(b,f)=C_{\mathrm{c}}(b,f)=2d$)、スケーリングを見積もる。($d=2$ の場合は図 5.3 参照) そうした場合の $\left|(b^L)^{-1}(f_{\min})\right|$ の変化を考えると、$\left|(b^L)^{-1}(f_{\min})\right|=|S_0|$ となる $L=L_0$ までの変化を予測することができる。$L=1,2,3$ については

$$\frac{\left|b^{-1}(f_{\min})\right|}{|S_{\min}|}\leq 1+2d \tag{5.1.9}$$

$$\begin{aligned}\frac{\left|(b^2)^{-1}(f_{\min})\right|}{|S_{\min}|}&\leq 1+2\binom{d}{1}+2\binom{d}{1}+4\binom{d}{2} \\ &=2d(d+1)+1\end{aligned} \tag{5.1.10}$$

$$\frac{\left|(b^3)^{-1}(f_{\min})\right|}{|S_{\min}|}\leq \frac{4}{3}d^3+2d^2+\frac{8}{3}d+1 \tag{5.1.11}$$

となる。すると $L=1,2$ の場合は、少なくとも式 (5.1.6) は成り立たない。実際 1 を越えるべきクエリ数の比が

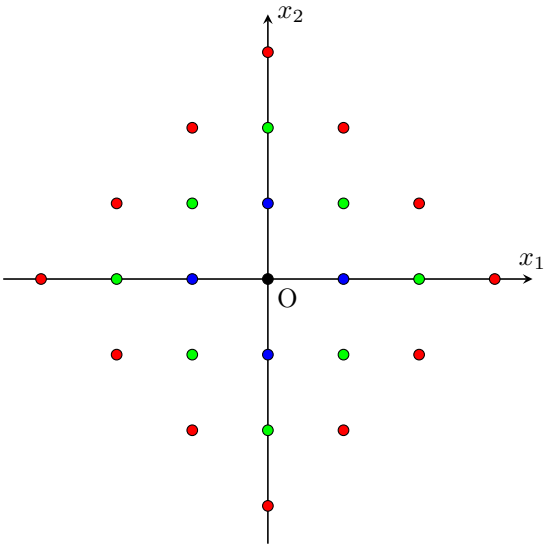


図 5.3: 局所探索の具体例 ($d=2$)

黒、緑、青、赤の順に局所的探索の回数 L を増やすことで探索される空間が増大する。

$$\begin{aligned}\frac{\sqrt{\frac{|S|}{|S_{\min}|}}}{(1+2LC_{\mathrm{q}}(b,f))\sqrt{\frac{|S|}{|(b^L)^{-1}(f_{\min})|}}}&\leq\begin{cases}\frac{\sqrt{1+2d}}{4d+1} & (L=1)\\ \frac{\sqrt{1+2d(d+1)}}{8d+1} & (L=2)\end{cases}\\ &\leq 1\end{aligned} \tag{5.1.12}$$

と 1 を超えない。より一般の L_0 未満の L について

$$\frac{\left|(b^L)^{-1}(f_{\min})\right|}{|S_{\min}|}\leq 1+\sum_{l=1}^{L}\sum_{k=1}^{\min(d,l)}\binom{d}{d-k}\binom{l-1}{k-1}2^k=1+2d\sum_{l=1}^{L}{}_2F_1\left(1-d,1-l;2;2\right) \tag{5.1.13}$$

である。ここで ${}_2F_1\left(a,b;c;z\right)$ は超幾何関数

$${}_2F_1\left(a,b;c;z\right)=\sum_{n=0}^{\infty}\frac{(a)_n(b)_n}{(c)_n}\frac{z^n}{n!} \tag{5.1.14}$$

、$(x)_n$ はポッホハマー記号

$$(x)_n=\begin{cases}1 & (n=0)\\ x(x+1)\cdots(x+n-1) & (n>0)\end{cases} \tag{5.1.15}$$

である。これは、d 次元空間における錐体の体積と x_i の符号の取り方を考えることで抑えることができて

$$\frac{(2L)^d}{d!}\leq 1+2d\sum_{l=1}^{L}{}_2F_1\left(1-d,1-l;2;2\right)\leq\frac{\left(2(L+\frac{d}{2})\right)^d}{d!} \tag{5.1.16}$$

が成り立つ。

同時に ${}_2F_1\left(1-x,1-x;2;2\right)$ は $x\in\{1,2,\ldots,100\}$ において近似的に 0.0091×5.6^x とかけ、式 (5.1.13) は $d<100$ において上から

$$\begin{cases}O\left(dL\cdot 5.6^d\right) & (L<d)\\ O\left(d(L-d+1)\cdot 5.6^L\right) & (L\geq d)\end{cases}$$

によって抑えることができる。

以上の計算によって簡単な例であればいくつかの仮定のもと、特定の局所探索における適切な更新回数 L の設定を行うことができる[*4]。

また具体的な量子的 multi-start 法では少なくとも $L > 2$ が必須であったが、古典的には $L > 1$ であることが必須とされる。これをより一般化してもし古典的 multi-start 法で反復回数 L の局所探索を行うことで pure random search よりも効率よく探索できる、つまり

$$\frac{(1+LC)\frac{|S|}{|(b^L)^{-1}(f_{\min})|}}{\frac{|S|}{|S_{\min}|}} = O(1) \tag{5.1.17}$$

だとしても、同じ回数だけ古典的局所探索 ($C_{\mathrm{q}} = C_{\mathrm{c}}$) による更新を行った量子的 multi-start 法では、

$$\frac{(1+2LC)\sqrt{\frac{|S|}{|(b^L)^{-1}(f_{\min})|}}}{\sqrt{\frac{|S|}{|S_{\min}|}}} = O\left(\frac{1+2LC}{\sqrt{1+LC}}\right) \tag{5.1.18}$$

であって、量子的に全探索を行った場合よりも効率が良いかはわからない。これは振幅増幅と局所探索法によりそれぞれ異なる加速を与え問題を複雑化しているからである。このように古典・量子的な multi-start 法では局所探索による加速と最小値探索による量子加速の影響によって、やはり局所探索 b や更新回数 L の設定は非自明である。

5.2 量子的 multi-start 法

前節の議論を、一般の局所最適化を用いた、量子振幅増幅による関数の最小化のための確率的アルゴリズムへと拡張する。まず確率的挙動を与えるために、疑似乱数の表に相当する集合 R の要素 $r \in R$ を疑似乱数のシード値とする。次に最適化において過去の情報を利用し行う可能性も含めて、ある集合 H の要素 $h \in H$ を最適化の過去の履歴の、完全なもしくは不完全な情報を表すとする。そしてクエリコスト評価のために、$c \in \mathbb{Z}_{\geq 0}$ は局所最適化の実行に必要なクエリコストを表すとする。また遺伝的アルゴリズム（付録 C 参照）といった、多点探索による局所探索も存在するため、$0 \leq k \leq |S|$ なる整数 k と S の冪集合 $\mathcal{P}(S)$ に対し、部分集合 $\mathcal{P}_k(S) = \{p \in \mathcal{P}(S) \mid |p| = k\}$ を定義する。$\mathcal{P}_k(S)$ の要素を、k 個の点を初期状態における複数の解の候補として選ぶ際に用いる。

そして局所最適化アルゴリズムに対し $p \in \mathcal{P}_k(S), r \in R, h \in H, c \in \mathbb{Z}_{\geq 0}$ に対し、$x' \in S$ を確率変数としてもつ確率分布

$$P(x'; p, r, h, c) \tag{5.2.1}$$

を定義する。確率分布であるための条件として、$P(x'; p, r, h, c) \geq 0, \sum_{x' \in S} P(x'; p, r, h, c) = 1$ とする。以上の設定の上で、一般の局所最適化アルゴリズムを「複数の k 個の解の候補 p から最適化を開始し、最適化の履歴に関する情報 h に基づき確率 $P(x'; p, r, h, c)$ でクエリコスト c を用いて x' へ解を更新する」ものと定式化することができる。この定式化のもとで議論を進めることができるが、簡単のために初期状態における解の候補は $k = 1$ 個とする。そして $x \in S, r \in R, h \in H, c \in \mathbb{Z}_{\geq 0}$ に対し、$x' \in S$ を確率変数としてもつ確率分布

$$P(x'; x, r, h, c) \tag{5.2.2}$$

を定義し、局所的最適化アルゴリズムを「解の候補 $x \in S$ から最適化を開始し、最適化の履歴に関する情報 h に基づき確率 $P(x'; x, r, h, c)$ でクエリコスト c を用いて x' へ解を更新する」ものとする。

[*4] ここでは隣接格子点と比較するような特定の山登り法によって得られる最大の探索の効率化をみてきたが、これはあくまでこの局所探索手法が有効である場合のみに限る。例として $\frac{|(b^L)^{-1}(f_{\min})|}{|S_{\min}|} \approx 1 + 2d \sum_{l=1}^{L} {}_2F_1(1-d, 1-l; 2; 2)$ とすることができるのは、例えば大域的最小点を中心にして単峰性を持つような、特定の場合である。

今まで見たきたように、Pure random search によって最小解が得られる確率は $\frac{|S_{\min}|}{|S|}$ であった。その確率は、初期解 x と疑似乱数のシード値 r に関する一様ランダムな選択をしたのち、コスト c のクエリと履歴 h に基づく局所最適化アルゴリズムをすることで、

$$(\text{局所最適化後に最小値が得られる確率}) = \frac{1}{|R||S|} \sum_{\substack{x,x' \in S, r \in R \\ f(x')=f_{\min}}} P(x';x,r,h,c) \tag{5.2.3}$$

へと変化する。

この局所最適化アルゴリズムを含んだ量子的 multi-start 法を定式化する。量子状態の確率振幅をアルゴリズムの確率的な過程の導入に用いる。履歴 h とクエリコスト c を用いる局所最適化に対応するユニタリ変換 $U_b(h,c)$ を、ある解の候補 x、自然数 r に関する計算基底状態 $|x\rangle, |r\rangle$ として、以下

$$U_b(h,c)\frac{1}{\sqrt{|R||S|}} |x\rangle |r\rangle |0\rangle |0\rangle = \sum_{x' \in S} \sqrt{P(x';x,r,h,c)}\, |x\rangle |r\rangle |h'\rangle |x'\rangle \tag{5.2.4}$$

を満たすとする。局所最適化により得られた新しい履歴を $|h'\rangle$ としている。すると古典的手法における「初期解 x と疑似乱数のシード値 r に関する一様ランダムな選択」に対応するような、量子状態の初期化は

$$\frac{1}{\sqrt{|R||S|}} \sum_{x \in S, r \in R} |x\rangle |r\rangle |0\rangle |0\rangle |0\rangle \tag{5.2.5}$$

である。この初期状態に対し、局所最適化を行うということは、ユニタリ変換 $U_b(h,c)$ を左から順に第 1 から第 4 番目のレジスタに作用させることに相当し、状態は

$$\frac{1}{\sqrt{|R||S|}} \sum_{x,x' \in S, r \in R} \sqrt{P(x';x,r,h,c)}\, |x\rangle |r\rangle |h'\rangle |x'\rangle |0\rangle \tag{5.2.6}$$

と変化する。新しい解の候補 x' に対する関数値 f のクエリが必要なので、ユニタリ変換 U_f を $U_f |x\rangle |0\rangle = |x\rangle |f(x)\rangle$ としたものを用いる。U_f を左から第４、５番目のレジスタに作用することで、

$$\frac{1}{\sqrt{|R||S|}} \sum_{x,x' \in S, r \in R} \sqrt{P(x';x,r,h,c)}\, |x\rangle |r\rangle |h'\rangle |x'\rangle |f(x')\rangle \tag{5.2.7}$$

は「初期解 x と疑似乱数のシード値 r に関する一様ランダムな選択」をし、局所最適化を行う古典的 multi-start アルゴリズムと一致する。なぜなら、この状態を x, x', r の計算基底状態で射影測定した時に最小値が得られる確率は、式 (5.2.3) と一致するためである。そしてあるクエリコスト c、履歴 h に基づく、$|0\rangle$ から状態 (5.2.7) への変換を $\mathcal{A}(h,c)$ として

$$\mathcal{A}(h,c)|\mathbf{0}\rangle \equiv \frac{1}{\sqrt{|R||S|}} \sum_{x,x' \in S, r \in R} \sqrt{P(x';x,r,h,c)}\, |x\rangle |r\rangle |h'\rangle |x'\rangle |f(x')\rangle \tag{5.2.8}$$

を満たすものとする[*5]。

変換 $\mathcal{A}(h,c)$ に対し最小値探索量子アルゴリズムを走らせれば、量子的 multi-start 法が前節と同じく可能になる[*6]。そのアルゴリズムを **Algorithm 7** に示す。停止条件としてあらかじめ可能な計算資源により M を定める。そして観測に応じて局所最適化手法を変更・調整する場合は、ステップ 25 のようにパラメータ m を定める l のリセットを行う。過去に同じ手法を用いていた場合はステップ 22 において結果を再利用している。そうした発見法的な場合に対しては、手法を変更しない段階においてサンプリングのみに関して優位性を証明できる[*7]。以下簡単にその証明を与える。（文献 [59] の付録において一般化した最小値探索について一部証明が示されている）

[*5] $\mathcal{A}(h,c)|\mathbf{0}\rangle$ の各成分に対し位相の自由度は存在する。

[*6] 正確には、最小値探索量子アルゴリズムが適用可能であるのは解の割合が全体の 3/4 以下である時であるので、量子振幅増幅アルゴリズムのパラメーター更新方法のように修正する。

[*7] 発見法的手法の改善を含んだアルゴリズム全体として証明をすることは困難。

Algorithm 7 Quantum multi-start algorithm

S 上で一様ランダムに解の候補 x_1 を生成し、$y_1 = f(x_1)$ とする。
λ を $1 < \lambda < 2$ なるある定数とする。
$h_1 = \varnothing$、c_1 をある整数 $c \in \mathbb{Z}_{\geq 0}$ とする。
$P(x'; p, r, h_1, c_1) \equiv P_1$ を定める。
$n = 1, l_0 = 0$ とする。
while f の全クエリ回数が M を超えない **do**
    $l_n = l_{n-1} + 1, m = \lceil \lambda^{l_n} \rceil$ とする。
    $\mathcal{A}(h_n, c_n)\,|0\rangle$ を測定し、解の候補を x、対応する値を $y = f(x)$ とする。観測された新たな履歴を h とする。
    if $y < y_n$ **then**
        $x_{n+1} = x,\ y_{n+1} = y$
    else
        $\{1, 2, \ldots, \lceil m \rceil\}$ から一様ランダムに整数 j を取り出す。
        $\mathcal{A}(h_n, c_n)$ に対し、閾値 y_n 未満であることを増幅条件として j 回の振幅増幅された $\mathbf{Q}^j \mathcal{A}\,|0\rangle$ の出力を測定し、解の候補を x'、対応する値を $y' = f(x')$、観測された新たな履歴を h' とする。
        if $y' < y_n$ **then**
            $x_{n+1} = x',\ y_{n+1} = y'$
        else
            $x_{n+1} = x_n,\ y_{n+1} = y_n$
        end if
    end if
    新たな履歴 h_{n+1} を履歴 $h_n,\ h,\ h'$ から生成する。
    履歴 h_{n+1} から $c_{n+1},\ P(x'; p, r, h_{n+1}, c_{n+1}) \equiv P_{n+1}$ を設定する。　▷ choose local optimization method
    if $c_{n+1} = c_k$ かつ $P_{n+1} = P_k$ なる最大の n 以下の k が存在する。 **then**
        $l_n = l_k$ とする。
    else
        $l_n = 0$ とする。　▷ reset parameter l_n
    end if
    $n \leftarrow n + 1$
end while

定理 5.2.1. 量子的 *multi-start* 法：確率的局所最適化に対する *multi-start* 法の高速化

（変動しうる）クエリコスト c を要する確率的局所最適化手法によって関数の最小化を考える。古典的 *multi-start* 法によって「複数の初期解の候補と疑似乱数のシード値 r を一様ランダムに選択して確率的局所最適化を行う」。それにより最小値がある 0 でない確率 p_1 で得られるとする。同手法を量子回路内に観測を含まない形で少なくとも古典計算と同程度に効率的に再現できるとする。

観測に応じて確率的局所最適化手法を変化させない[*8]場合、古典的には $O\left({}^{\max(1,c)}/_{p_1}\right)$ のクエリ数の期待値で最小値が得られるが、***Algorithm 7*** によりクエリ数の期待値 $O\left({}^{\max(1,c)}/_{\sqrt{p_1}}\right)$ によって最小値が得られる。

観測に応じて確率的局所最適化手法を変化させる場合、手法を変更しない間に最小値が得られるまでのクエリ数の期待値は、手法ごとに再定義したクエリコスト c と確率 p_1 を用いて、古典的には $O\left({}^{\max(1,c)}/_{p_1}\right)$、量子的

には $O\left({}^{\max(1,c)}/_{\sqrt{p_1}}\right)$ となる。

証明. ここまでの説明と同じく初期解の数は簡単のために 1 つであるとする。そして証明の途中まで $c \equiv c_1 = c_2 = \ldots$, $P \equiv P_1 = P_2 = \ldots$ を仮定する。

$f(S)$ のうち i $(= 1, 2, \ldots, |f(S)|)$ 番目に小さな数を f_i とおく。局所最適化後に値 f_i が得られる確率を

$$\pi_i \equiv \frac{1}{|R||S|} \sum_{\substack{x,x' \in S, r \in R \\ f(x') = f_i}} P(x'; x, r, h, c) \tag{5.2.9}$$

とおく。ただし、$\sum_{i=1}^{|f(S)|} \pi_i = 1$ を満たし、仮定から $\pi_1 > 0$ とする。また f_i 以下の値が局所最適化後に得られる確率を $p_i \equiv \sum_{j=1}^{i} \pi_j \equiv \sin^2 \theta_i > 0$ とおく。ただし $0 \leq \theta_i \leq {}^{\pi}/_{2}$ とする。この時 **Algorithm 7** における閾値の更新方法は strong pure adaptive search[*9]によるものであるから、文献 [60] によりアルゴリズムにおいて f_i が現れる確率は、

$$\frac{\pi_i}{p_i} \tag{5.2.10}$$

と表される。すると、最小値 f_1 が得られるまでの **Algorithm 7** における $\mathcal{A}$ の実行回数の期待値は、

$$\sum_{i=2}^{|f(S)|} \frac{\pi_i}{p_i} \times (f_i\text{が得られているときの、}f_{i-1}\text{以下の } f \text{ の値が得られるまでの } \mathcal{A} \text{ の実行回数の期待値}) \tag{5.2.11}$$

によって上から抑えられる。古典的には同式は

$$\sum_{i=2}^{|f(S)|} \frac{\pi_i}{p_i} \frac{1}{p_{i-1}} \leq \frac{\pi_2}{p_1 + \pi_2} \frac{1}{p_1} + \sum_{i=3}^{|f(S)|} \frac{1}{p_{i-1}^2} \leq \frac{1}{p_1} + \frac{1}{p_1} = O\left(\frac{1}{p_1}\right) \tag{5.2.12}$$

から $\mathcal{A}$ の実行回数の期待値は、$O({}^{1}/_{p_1})$。

Algorithm 7 の場合を考える。量子振幅増幅における **Algorithm 1** における定理 2.4.2 の証明から、**Algorithm 7** において f_i が得られているときの、f_{i-1} 以下の f の値が得られるまでの $\mathcal{A}$ の実行回数の期待値は $O\left({}^{1}/_{\sin\theta_{i-1}}\right) = O\left({}^{1}/_{\sqrt{p_{i-1}}}\right)$ となる。故に式 (5.2.11) は

$$O\left(\sum_{i=2}^{|f(S)|} \frac{\pi_i}{p_i} \frac{1}{\sqrt{p_{i-1}}}\right) \tag{5.2.13}$$

であって、

$$\sum_{i=2}^{|f(S)|} \frac{\pi_i}{p_i} \frac{1}{\sqrt{p_{i-1}}} = \frac{\pi_2}{p_1 + \pi_2} \frac{1}{\sqrt{p_1}} + \sum_{i=3}^{|f(S)|} \frac{1}{(p_{i-1})^{\frac{3}{2}}} \leq \frac{1}{\sqrt{p_1}} + \frac{2}{\sqrt{p_1}} = O\left(\frac{1}{\sqrt{p_1}}\right) \tag{5.2.14}$$

である。

もし $c_{n+1} \neq c_n$ または $P_{n+1} \neq P_n$ なる自然数 n が存在していたとしても、少なくとも $\mathcal{A}$ の実行回数の期待値を上から抑えることに関しては、上と全く同じ議論が成り立つ。つまりクエリコスト c または確率分布 P を変化させた後に関数 f の最小値が得られるまでの古典的・量子的 $\mathcal{A}$ の実行回数すなわち f のクエリ数の期待値は、再定義した p_1 を用いて式 (5.2.12), 式 (5.2.14) で抑えられる。

最後に $\mathcal{A}, \mathcal{A}^{-1}$, 増幅のための判定は f のクエリコスト $c, c, 1$ を要することを考慮すれば、f のクエリコストの期待値はオーダーとして $\max(1, c)$ に比例し証明が終わる。 □

[*8] ここではアルゴリズム内においてクエリコスト c または確率分布 P を変化させた場合、局所最適化手法を変更すると呼んでいる。

[*9] 閾値の条件として等号を含めないものを strong、そうでないものを weak な PAS と呼ぶ。

λ の設定方法に関しては文献 [61] によれば、DH アルゴリズムにおいて $\lambda = 1.34$ とすることで数値計算により $\mathcal{A}$ の実行回数の期待値の上限が最も抑えられることが示されている[*10]。今の場合仮に局所最適化手法を変えないのであれば、$\mathcal{A}$ 及び $\mathcal{A}^{-1}$ の実行回数の期待値 $F(\lambda, \theta)$ は

$$F(\lambda,\theta) = \sum_{l=1}^{\infty} \left\{ \sin^2\theta \cdot 1 + \left(1-\sin^2\theta\right) \cdot \left(1 + 2\frac{1+\lceil\lambda^l\rceil}{2}\right) \right\} \alpha_l \tag{5.2.15}$$

$$\alpha_l \equiv \prod_{j=1}^{l-1} \left(1-\sin^2\theta\right) \left\{ \frac{1}{2} + \frac{1}{2\lceil\lambda^j\rceil} \left(\frac{\sin\left(4\theta\left(1+\lceil\lambda^j\rceil\right)\right)}{2\sin(2\theta)} - \cos(2\theta) \right) \right\} \tag{5.2.16}$$

とかける。ただし最小値の全体に占める割合を $\sin^2\theta \equiv p_1$ とした。ここまでの証明からこれが $\Theta\left({}^1/_{\sin\theta}\right)$ であることは分かっている。アルゴリズムを λ に関して最適化する一つの方法は、この無限級数 $F(\lambda,\theta)$ を ${}^1/_{\sin\theta}$ で割った量の $0 \le \theta \le {}^\pi/_2$ における最大値 $\max\limits_{0\le\theta\le{}^\pi/_2} (\sin\theta F(\lambda,\theta))$ が最小となるように λ を選択することである。実際に部分和を数値計算をすると $\lambda \approx 1.30$ が最良であることがわかる。

Algorithm 7 について補足する。前節と同じく、この局所最適化手法を用いた量子的 multi-start 法が量子的全探索に比べクエリ数に関して効率的でなければならないという問題が存在する。それに対する対処法としては、あるクエリコストまでは局所最適化手法が少なくとも有効であるという仮定をし、確率分布 $P(x';p,r,h,c)$ で決定づけられる局所最適化アルゴリズムの動作として、解の更新がされなくなった場合は新たにサンプリング点をとり局所最適化を行うものを構成すればいい。これによりクエリの “無駄遣い” を避けることができ、一定程度までは問題が解消される。しかし結局それは、量子的 multi-start 法の内部で、古典的 multi-start 法を行うことになるので、可能である限り多数回更新が止まるような状況は避けるようにクエリコスト c を設定するべきである。しかしそれはやはり発見法的な手法か、何らかの仮説に基づき行うしかない。

逆に局所最適化の途中で観測される可能性もあるため、振幅増幅後に再度増幅なしに局所的最適化を行うことも考えられる。しかし、観測後に必要な最適化コストが増えれば増えるほど、量子的 multi-start 法による加速が低下することに注意する必要がある。

そうした適切な c の設定に関する仮説として d を目的関数の次元として $c = O\left(\log^d |S|\right)$ が考えられる。なぜなら文献 [62] において、DH の最小値探索アルゴリズムにおける増幅した状態を観測した後に、振幅増幅なしに古典的最適化アルゴリズムを走らせ更新が止まった場合その時の f の値を閾値として更新することが考えられているが、その際古典的最適化アルゴリズムによって $c = O\left(\log^d |S|\right)$ のクエリが必要になるとされている。$c = O\left(\log^d |S|\right)$ が十分であるかについて、整列されたデータ内での探索は二分探索によりデータ数に対し対数のクエリが必要になることや、$|S|$ 自体が d に関し指数的であるから $c = O\left(d^d\right)$ を考慮すると、一定程度の問題においては十分なクエリコストであると解釈できる。以上の解釈も踏まえ、文献に倣って $c = \Theta\left(\log^d |S|\right)$ と仮定する。この c を用いて multi-start 法の古典的なものに対する加速を評価することができる。

評価にあたって、そもそも multi-start 法が全探索手法に比べクエリ数に関して効率的でなければならないという前提が存在する。それは古典的には確率 p_1（式 (5.2.9) 参照）を用いて条件

$$r_{\mathrm{c}} \equiv \frac{\max\left(1, c_{\mathrm{c}}\right)}{\frac{p_1}{\frac{|S|}{|S_{\min}|}}} < (\text{constant}) \tag{5.2.17}$$

となる（$\max\left(1, c_{\mathrm{c}}\right)$ は局所探索自体のコスト c_{c} と最適化後の解の候補に対するクエリコスト 1 に由来する）が、こ

[*10] 最小値の全体に占める割合を $\sin^2\theta \equiv p_1$ とした時、DH アルゴリズムのクエリの期待値は $\sum\limits_{j=0}^{\infty} \frac{\lceil\lambda^j\rceil}{2} \prod\limits_{i=0}^{j-1} \left(\frac{1}{2} + \frac{\sin\left(4\theta\lceil\lambda^i\rceil\right)}{4\lceil\lambda^i\rceil\sin(2\theta)}\right)$。DH アルゴリズムを λ に関して最適化するには、この無限級数を ${}^1/_{\sin\theta}$ で割った量の $0 \le \theta \le {}^\pi/_2$ における最大値が最小となるように λ を選択すればいい。

こでは量子 multi-start 法が量子最小値探索に比べ効率的である必要がありそれは

$$r_{\mathrm{q}} \equiv \frac{\max\left(1, c_{\mathrm{q}}\right)}{\sqrt{p_1}\sqrt{\frac{|S|}{|S_{\min}|}}} < (\text{constant}) \tag{5.2.18}$$

となる。$c_{\mathrm{c}} \geq c_{\mathrm{q}}$ を仮定し、$c_{\mathrm{q}}, c_{\mathrm{c}} \geq 0, |S_{\min}| \geq 1$ から古典、量子 multi-start に対するクエリの期待値を

$$\frac{\max\left(1, c_{\mathrm{c}}\right)}{p_1} = O\left(r_{\mathrm{c}}|S|\right) \tag{5.2.19}$$

$$\frac{\max\left(1, c_{\mathrm{q}}\right)}{\sqrt{p_1}} = \max\left(1, c_{\mathrm{q}}\right)\sqrt{\frac{r_{\mathrm{c}}|S|}{\max\left(1, c_{\mathrm{c}}\right)|S_{\min}|}} = O\left(\sqrt{\max\left(1, c_{\mathrm{q}}\right) r_{\mathrm{c}}|S|}\right) \tag{5.2.20}$$

と表せる。$c_{\mathrm{c}} = c_{\mathrm{q}} = \Theta\left(\log^d |S|\right)$ として d を固定した定数として見た場合、$\max\left(1, c_{\mathrm{q}}\right)/\sqrt{p_1} = O\left(\log^{\frac{d}{2}} |S|\sqrt{r_{\mathrm{c}}|S|}\right) = \widetilde{O}\left(\sqrt{r_{\mathrm{c}}|S|}\right)$ となる。以上の考察により、量子的 multi-start 法において必要なクエリ数を、もし古典的 multi-start 法が有効（$r_{\mathrm{c}} = O\left(1\right)$）であれば $c = O\left(\log^d |S|\right)$ という現実的設定において、$\widetilde{O}\left(\sqrt{|S|}\right)$ であるという上限を与えることができる。これが量子最小値探索を上回るか否かは $O\left(r_{\mathrm{c}}|S|\right)$ の漸近的振る舞いが決定していて問題の構造と手法に依存するが、対数因子を無視すれば少なくとも量子最小値探索と同程度の計算量で済む事がわかり、一つの重要な結果が導かれた。

観測を含まない局所最適化アルゴリズム自体に対しても量子加速が可能であったとしても、クエリコスト c に対し量子加速が指数的（$|S|$ に関して分数指数または多項式以上）でなければいずれにせよ加速の因子は $\widetilde{O}\left(\sqrt{|S|}\right)$ と変わらない。また最適化の途中において局所最適化手法を履歴に基づき変更したとしてもこの評価が成り立つ。以上の議論の結果を定理にまとめる。

定理 5.2.2. 量子的 *multi-start* 法の古典的手法に対する加速の評価

$O\left(\log^d |S|\right)$ のクエリコストを要する確率的局所最適化アルゴリズムを用いた古典的 *multi-start* 法が有効であり、かつ局所最適化アルゴリズム自体の量子加速が $|S|$ に関して高々対数とする。

この時古典的 *multi-start* 法と量子的 *multi-start* 法による最小化に必要なクエリ数はそれぞれ $O\left(|S|\right)$, $\widetilde{O}\left(\sqrt{|S|}\right)$ である。それぞれのより強い上限は、個別の局所最適化手法ごとに定義された有効性の指標 r_c を用い $|S|$ の対数因子を許して $O\left(r_c|S|\right), \widetilde{O}\left(\sqrt{r_c|S|}\right)$ と与えられる。

さらなる補足として確率振幅として $P(x'; p, r, h, c)$ を量子回路内で効率的に構成できるかという問題がある。この問題はつまり古典的なオプティマイザーに対応するものを観測を含まない量子回路内で効率的に構築できるかという問いと、ガウス分布といった確率分布関数を量子状態の振幅として効率的にエンコードできるかという二つの問いに分解できる。前者に関しては最適化問題全般に関わる問題であり、本論文では勾配推定に関してのみ触れる。後者に関しては、離散確率分布関数 $p_N : \{0, \ldots, N-1\} \to \{0, 1\}$, $\Sigma_x p_N(x) = 1$ が与えられた時に、状態

$$|\Psi\rangle = \sum_{x=0}^{N-1} \sqrt{p(x)} e^{2\pi i \psi(x)} |x\rangle \tag{5.2.21}$$

を、近似的にだが量子ビット数 $\log_2 N$ に関して効率的に構成できることが知られている [63]。より詳細に述べると、ある η に対し $p_N(x) \leq 1/\eta N$ for all x and N であれば、任意の十分小さな正の数 ν, λ に対し $\langle\widetilde{\Psi}|\Psi\rangle > 1 - \lambda$ なる $|\widetilde{\Psi}\rangle$ を確率 $1 - \nu$ 以上で、$\nu^{-1}, \lambda^{-1}, \eta^{-1}$ に関して多項式時間で用意することができる。

他にもこうした離散確率分布を効率的に振幅に符号化する手法として、一次元の積分可能な確率密度関数を対象とした Grover-Rudolph アルゴリズム [64] が著名だが、それを大幅に改善した近年の結果として、ある種の滑らかさを

仮定した上で漸近的に量子ビット数によらずに状態準備を可能とする手法が提案されている [65]。よって、少なくともある種の確率分布に従って局所的最適化を行うアルゴリズムは、量子回路内で効率的に実現できるであろうと考えられる。

定理に関する最も重要な補足として、構成の上では履歴を用いて局所最適化手法を変化させていくことが可能としたが、**Algorithm 7** はサンプリングの部分でしか量子加速を与えないため、最適化のステップごとに局所最適化手法を変更すればするほど量子優位性は失われ、仮に毎ステップ履歴に基づき局所最適化を変更すれば全く優位性はなくなる。文献 [66] では、そうした量子振幅増幅の問題点について指摘されていて、履歴に基づくことの多いヒューリスティックスについてはその問題特有のアルゴリズムを作る必要があるとされる。そもそもヒューリスティックスは個々の探索手法の変更に依存する複雑なものであるため、理論的な解析は難しいとされる。そのため困難とされる特定の問題に対して効力を発揮するかどうかによってその性能が評価されるが、主にそうした問題として NP 完全[*11]な問題が選ばれる。結局そうしたクラスの問題またはその特殊な場合を量子計算によって効率よく解けるかということになり、現在も研究が続けられている。

一般的な定理 5.2.1, 定理 5.2.2 の結果によって、量子勾配推定と量子振幅増幅を組み合わせた時の量子加速について議論することができる。ここでは Jordan のアルゴリズムを解析し、量子勾配推定に関してより一般化した結果を与える文献 [17] の結果を主にして議論をする。それによると滑らかな関数[*12]に対し、関数の次元が d であったときに L^∞ ノルムにおいて精度 ϵ で勾配を求めることを考える。古典計算では $\widetilde{O}\left(d^2/\epsilon\right)$ のクエリが必要だが、量子勾配推定によって $\widetilde{O}\left(d/\sqrt{\epsilon}\right)$ で求めることが可能である。これは 2 次の量子加速であるが Grover 探索による加速ではなく、同時に原理的に観測を含まないアルゴリズムであるため定理 5.2.2 の形式に落とし込むことができる。局所最適化として量子勾配推定によって得られた勾配をもとに、通常の勾配降下法により平均的に T ステップの局所的最小化を行うとしたとき、古典計算では関数最小化に M のクエリが必要とする。定理 5.2.2 の仮定を用いて量子計算機によって $\widetilde{O}\left(\sqrt{M}\right)$ のクエリで十分であることが言える。ただしより正確には $\widetilde{O}\left(\sqrt{T\epsilon/\sqrt{d}M}\right)$ であって $M \propto T$ であるから反復回数 T に対しての加速は起きない。

このように量子的 multi-start 法という描像を用いて局所最適化を含んだアルゴリズムにおける探索またはサンプリングによる量子加速について、包括的に与えることができる。局所探索の一種である遺伝的アルゴリズムにおける量子計算の応用は既に考案されている [67] ように、サンプリング以外の部分で特定の構造を持った問題に対し量子加速を与えることができるか否かの研究がより一層重要であると思われる。

[*11] 非決定性チューリングマシンによって多項式時間で解ける決定問題のクラス (NP) に属し、かつ NP に属す任意の問題から多項式時間帰着が可能なものを NP 完全という。

[*12] 滑らかさの定義は原論文を参照されたい

第 6 章

まとめ

本研究では、主に最適化問題における最適化過程において量子計算が果たすと思われる役割について理解することを目的とした。より具体的な最適化を行う方法として、勾配法のための関数の勾配を効率的に推定する量子勾配推定と、効率的な探索手法である量子探索という 2 つの方法に注目した。前者に関しては、新たに位相の対称性を用い整数係数・整数変数多項式関数に対しビット表現に収まるように勾配の各成分が十分小さいと仮定した上で、関数の次数に関して定数という古典計算よりも効率的なクエリ計算量で勾配を推定可能であることを示した。そして考案したアルゴリズムを量子勾配推定の実機による検証としても初の試みと考えられるため実験を行い、部分的に検証することに成功した。後者に関しては、計算基底における数値計算を前提として、古典的機械学習モデルであるパーセプトロンと FNN の構築を行い、量子探索を用いた学習シミュレーションを行なった。この手法自体は 20 年以上前から存在していたが、特に FNN に関しては一から量子回路内で構築した先行研究は少ないと考えられる。また量子探索を、実用面を考慮し局所的構造を持った問題に対して適用した時へ一般化を行い、発見法的な手法の改善を含みつつサンプリングに関し量子加速をもたらす量子的 multi-start 法として定式化を行った。こうした実際上の問題を踏まえた定式化と定量的議論はなされていないと考えられる。そしてこの二つの視点に基づいた研究を通し、今後位相の対称性の利用に関連して量子系への情報の符号化や、サンプリング以外の実用的な問題の構造を適切に利用した量子計算に関する研究が引き続き重要な研究対象になり続けると考えられる。

謝辞

本研究を進めるのにあたって、多くの方にお世話になりました。ここに感謝の意を申し上げます。

指導教官である浅井祥二教授には、研究のみならず物事の考え方についてご指導いただきましたし、研究テーマの変更についても寛容に受け入れてくださいました。澤田龍准教授には、所属や研究会発表のための申請の各種手続きをしていただいたり研究の方向性に関して相談に乗っていただきました。寺師弘二准教授には、本研究や研究発表、研究会への投稿原稿を通して研究における問題点を鋭く指摘していただきました。飯山悠太郎助教には、最適化過程における量子計算の可能性という形で本テーマをいただきましたし、研究全般の助言をいただきました。永野廉人特任研究員には、定例ミーティングで質問や貴重なアドバイスをいただきました。そして ICEPP の量子コンピューティンググループの皆様には、日々議論に参加させていただきました。

浅井研究室および tabletop 実験グループを通し、難波俊雄助教、石田明助教、神谷好郎助教、稲田聡明特任助教、周健治さん、上岡修星さん、清野結大さん、成田佳奈香さん、橋立佳央理さんには、大学院で研究をするにあたって多くのご支援をいただきました。ICEPP 物理事務室のみなさまには大学院修士過程の生活を送る上で各種サポートをしていただきました。研究室同期の並木飛鳥さんや盧承佑さん、後輩の寺尾萌里乃さん、山下恵理香さん、量子コンピューティンググループにおける同期の大久保龍之介さんや張元豪さん、そして同期の村田樹さん、林雄一郎さん、吉田圭佑さん、藏嘉琦さん、青木匠さんまたは ICEPP 関係者の皆様には、研究の関係の有無を問わず日々仲良く接してくださいました。最後に、日々暖かく支えてくださった家族に感謝致します。

参考文献

[1] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[2] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5):1484–1509, Oct 1997.

[3] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science, pages 124–134. Ieee, 1994.

[4] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. Quantum annealing with manufactured spins. Nature, 473(7346):194–198, 2011.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. Nature, 574(7779):505–510, 2019.

[6] Ibm quantum, 2021.

[7] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554, 2006.

[8] Martin Hilbert and Priscila López. The world 's technological capacity to store, communicate, and compute information. science, 332(6025):60–65, 2011.

[9] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. Nature, 549(7671):195–202, 2017.

[10] John Preskill. Quantum computing in the nisq era and beyond. Quantum, 2:79, 2018.

[11] David P. DiVincenzo and Peter W. Shor. Fault-tolerant error correction with efficient quantum codes. Phys. Rev. Lett., 77:3260–3263, Oct 1996.

[12] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. Physical review A, 52(4):R2493, 1995.

[13] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. New Journal of Physics, 18(2):023023, 2016.

[14] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. Nature communications, 9(1):1–6, 2018.

[15] Bob Ricks and Dan Ventura. Training a quantum neural network. Advances in neural information processing systems, 16:1019–1026, 2003.

[16] Cesar Borisovich Pronin and Andrey Vladimirovich Ostroukh. Development and training of quantum neural networks, based on the principles of grover's algorithm. arXiv preprint arXiv:2110.01443, 2021.

[17] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1425–1444. SIAM, 2019.

[18] Patrick Rebentrost, Maria Schuld, Leonard Wossnig, Francesco Petruccione, and Seth Lloyd. Quantum gradient descent and newton's method for constrained polynomial optimization. New Journal of Physics, 21(7):073023, 2019.

[19] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. Quantum, 4:314, August 2020.

[20] Iordanis Kerenidis and Anupam Prakash. Quantum gradient descent for linear systems and least squares. Phys. Rev. A, 101:022316, Feb 2020.

[21] Pan Gao, Keren Li, Shijie Wei, Jiancun Gao, and Guilu Long. Quantum gradient algorithm for general polynomials. Physical Review A, 103(4):042403, 2021.

[22] David Bulger. Quantum basin hopping with gradient-based local optimisation. arXiv preprint quant-ph/0507193, 2005.

[23] David W Bulger. Combining a local search and grover ' s algorithm in black-box global optimization. Journal of optimization theory and applications, 133(3):289–301, 2007.

[24] Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics, 81(7):074001, 2018.

[25] Don Coppersmith. An approximate fourier transform useful in quantum factoring. arXiv preprint quant-ph/0201067, 2002.

[26] L. Hales and S. Hallgren. An improved quantum fourier transform algorithm and applications. In Proceedings 41st Annual Symposium on Foundations of Computer Science, pages 515–525, 2000.

[27] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. Mathematics of computation, 19(90):297–301, 1965.

[28] Michael Heideman, Don Johnson, and Charles Burrus. Gauss and the history of the fast fourier transform. IEEE ASSP Magazine, 1(4):14–21, 1984.

[29] Thomas G Draper. Addition on a quantum computer. arXiv preprint quant-ph/0008033, 2000.

[30] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. Contemporary Mathematics, 305:53–74, 2002.

[31] Lov K Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219, 1996.

[32] Lov K Grover. Quantum mechanics helps in searching for a needle in a haystack. Physical review letters, 79(2):325, 1997.

[33] Lov K Grover. Quantum computers can search rapidly by using almost any transformation. Physical Review Letters, 80(19):4329, 1998.

[34] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. SIAM journal on Computing, 26(5):1510–1523, 1997.

[35] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. arXiv preprint quant-ph/9511026, 1995.

[36] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 454(1969):339–354, 1998.

[37] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto. Amplitude estimation without phase estimation. Quantum Information Processing, 19(2):1–17, 2020.

[38] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. Fortschritte der Physik: Progress of Physics, 46(4-5):493–505, 1998.

[39] Christoph Durr and Peter Hoyer. A quantum algorithm for finding the minimum. arXiv preprint quant-ph/9607014, 1996.

[40] Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. Phys. Rev. Lett., 95:050501, Jul 2005.

[41] Albert B Novikoff. On convergence proofs for perceptrons. Technical report, STANFORD RESEARCH INST MENLO PARK CA, 1963.

[42] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum perceptron models. arXiv preprint arXiv:1602.04799, 2016.

[43] Ashley Montanaro. The quantum query complexity of learning multilinear polynomials, 2012.

[44] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 439(1907):553–558, 1992.

[45] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. SIAM Journal on computing, 26(5):1411–1473, 1997.

[46] Austin Gilliam, Stefan Woerner, and Constantin Gonciulea. Grover adaptive search for constrained polynomial binary optimization. Quantum, 5:428, 2021.

[47] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. Qiskit: An Open-source Framework for Quantum Computing, January 2019.

[48] Sergey Bravyi, Sarah Sheldon, Abhinav Kandala, David C. Mckay, and Jay M. Gambetta. Mitigating measurement errors in multiqubit experiments. Phys. Rev. A, 103:042605, Apr 2021.

[49] Andreas Griewank and Andrea Walther. Evaluating derivatives: principles and techniques of algorithmic differentiation. SIAM, 2008.

[50] Charles Audet and Warren Hare. Derivative-free and blackbox optimization. 2017.

[51] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. Physical review letters, 103(15):150502, 2009.

[52] Iordanis Kerenidis and Anupam Prakash. Quantum gradient descent for linear systems and least squares. Physical Review A, 101(2):022316, 2020.

[53] Laurent Meunier, Herilalaina Rakotoarison, Pak Kan Wong, Baptiste Roziere, Jeremy Rapin, Olivier Teytaud, Antoine Moreau, and Carola Doerr. Black-box optimization revisited: Improving algorithm selection wizards through massive benchmarking, 2021.

[54] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O ' brien. A variational eigenvalue solver on a photonic quantum processor. Nature communications, 5(1):1–7, 2014.

[55] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[56] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. Quantum Science and Technology, 2(4):045001, 2017.

[57] Le Baron O Ferguson. What can be approximated by polynomials with integer coefficients. The American Mathematical Monthly, 113(5):403–414, 2006.

[58] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pages 193–204, 2019.

[59] Joran Van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum sdp-solvers: Better upper and lower bounds. Quantum, 4:230, 2020.

[60] Zelda B Zabinsky and Robert L Smith. Pure adaptive search in global optimization. Mathematical programming, 53(1):323–338, 1992.

[61] William P Baritompa, David W Bulger, and Graham R Wood. Grover's quantum algorithm applied to global optimization. SIAM Journal on Optimization, 15(4):1170–1184, 2005.

[62] Pedro CS Lara, Renato Portugal, and Carlile Lavor. A new hybrid classical-quantum algorithm for continuous global optimization problems. Journal of Global Optimization, 60(2):317–331, 2014.

[63] Andrei N Soklakov and Rüdiger Schack. Efficient state preparation for a register of quantum bits. Physical review A, 73(1):012307, 2006.

[64] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. arXiv preprint quant-ph/0208112, 2002.

[65] Gabriel Marin-Sanchez, Javier Gonzalez-Conde, and Mikel Sanz. Quantum algorithms for approximate function loading, 2021.

[66] Tad Hogg. Quantum search heuristics. Physical Review A, 61(5):052311, 2000.

[67] Kuk-Hyun Han and Jong-Hwan Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512), volume 2, pages 1354–1360. IEEE, 2000.

[68] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. Quantum arithmetic with the quantum fourier transform. Quantum Information Processing, 16(6):152, 2017.

[69] Engin Şahin. Quantum arithmetic operations based on quantum fourier transform on signed integers. International Journal of Quantum Information, 18(06):2050035, 2020.

[70] Stuart Andrew Hadfield. Quantum algorithms for scientific computing and approximate optimization. Columbia University, 2018.

[71] Thomas Weise. Global optimization algorithms-theory and application. Self-Published Thomas Weise,

2009.

[72] Zelda B Zabinsky. Stochastic adaptive search for global optimization, volume 72. Springer Science & Business Media, 2003.

[73] David H Wolpert, William G Macready, et al. No free lunch theorems for search. Technical report, Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[74] David H Wolpert and William G Macready. No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1):67–82, 1997.

[75] Kyle Poland, Kerstin Beer, and Tobias J Osborne. No free lunch for quantum machine learning. arXiv preprint arXiv:2003.14103, 2020.

[76] Samuel H Brooks. A discussion of random methods for seeking maxima. Operations research, 6(2):244–251, 1958.

[77] Aleta Berk Finnila, MA Gomez, C Sebenik, Catherine Stenson, and Jimmie D Doll. Quantum annealing: A new method for minimizing multidimensional functions. Chemical physics letters, 219(5-6):343–348, 1994.

[78] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. Physical Review E, 58(5):5355, 1998.

[79] Rafael Martí. Multi-start methods. In Handbook of metaheuristics, pages 355–368. Springer, 2003.

付録 A

計算量の記法

計算量を評価する際に用いられるランダウの記号についてここで説明する。ランダウの記号によって関数の極限における漸近的振る舞いを記述することができる。いくつかの流儀が存在するが本論文では以下で定義する記号のみを用いることにする。

2つの実数値関数 $f,\ g\colon \mathbb{R} \to \mathbb{R}$ が与えられているとする。

定義 A.0.1.

$$f(x) = O(g(x)) \quad (x \to \infty) \tag{A.0.1}$$

を

$$\exists M > 0\ \exists x_0\ \forall x > x_0\ f(x) \leqq Mg(x) \tag{A.0.2}$$

を表すとして定義し、“$f(x)$ が $x \to \infty$ の時オーダー $O(g(x))$ である”という。

また Ω, Θ を用いたオーダー記法を定義する。

定義 A.0.2.

$$f(x) = \Omega(g(x)) \quad (x \to \infty) \tag{A.0.3}$$

を

$$\exists M > 0\ \exists x_0\ \forall x > x_0\ f(x) \geqq Mg(x) \tag{A.0.4}$$

を表すとして定義し、“$f(x)$ が $x \to \infty$ の時オーダー $\Omega(g(x))$ である”という。

定義 A.0.3.

$$f(x) = \Theta(g(x)) \quad (x \to \infty) \tag{A.0.5}$$

を

$$\exists M_1 > 0\ \exists M_2 > 0\ \exists x_0 > 0\ \forall x > x_0\ M_1 g(x) \leqq f(x) \leqq M_2 g(x) \tag{A.0.6}$$

を表すとして定義し、“$f(x)$ が $x \to \infty$ の時オーダー $\Theta(g(x))$ である”という。

オーダー評価における対数因子の差異は些細なものとして扱われるのでそして O, Θ, Ω に加えて˜を用いた記法を以下のように導入する。

定義 A.0.4.

$$f(x) = \widetilde{O}(g(x)) \quad (x \to \infty) \tag{A.0.7}$$

を

$$\exists k \ such\ that\ f(x) = O(g(x)\log^k(g(x))) \tag{A.0.8}$$

を表すとして定義し、“$f(x)$ が $x \to \infty$ の時オーダー $\widetilde{O}(g(x))$ である” という。

Ω, Θ に対しても $\widetilde{\Omega}$, $\widetilde{\Theta}$ が同様に定義されているとする。

多変数関数の漸近的振る舞いの記述への拡張も可能であるがここでは省く。関数のある座標点近傍における振る舞いを記述する際にもランダウの記法を用いることができる。

定義 A.0.5. $a \in \mathbb{R}$ に対し

$$f(x) = O(g(x)) \quad (x \to a) \tag{A.0.9}$$

を

$$\exists \delta > 0\ \exists M > 0\ such\ that\ \forall x, |x-a| < \delta \Rightarrow |f(x)| < M|g(x)| \tag{A.0.10}$$

を表すとして定義し、“$f(x)$ が $x \to a$ の時オーダー $O(g(x))$ である” という。

これに倣い座標点近傍における他のオーダー記法も極限 $x \to \infty$ と同様に定義しているとする。より一般にオーダー記法を関数の絶対値に関して導入することもでき、必要に応じて用いるとする。

計算量の記法を定義したので、いくつか主要な計算量とそれに属するアルゴリズムの例について表 A.1 にまとめておく。

記法	名称	アルゴリズム
$O(1)$	Constant time（定数時間）	Bernstein-Vazirani, Deutsch-Joza, Jordan
$O(\log n)$	Logarithmic time（対数時間）	二分探索、ユークリッドの互除法
$O(n^c),\ 0 < \exists c < 1$	Fractional power（分数指数関数）	Grover
$O(n)$	Linear time（線形時間）	加減算
$O(n \log n)$	Linearitimic（準線形、線形対数）	AQFT, ソート
$O(n^2)$	Quadratic time（二乗時間）	QFT, 乗除算
$O(n^c),\ \exists c \geqq 1$	Polynomial time（多項式時間）	カーマーカーのアルゴリズム[*1]
$O(2^n)$	Exponential time（指数時間）	充足可能性問題における全探索
$O(n!)$	Factorial, Combinational（階乗関数）	巡回セールスマン問題における全探索

表 A.1: オーダー記法による分類とそれに属するアルゴリズム

[*1] 線形計画問題に対する初の実用的なアルゴリズムとして有名である。

付録 B

数の表現と初等演算

B.1 固定小数点数法による数の表現

一般に数の表現方法として、浮動小数点数形式と固定小数点数形式が存在する。量子アルゴリズムにおいては固定小数点数形式を用いることが多い。そのため以下では浮動小数点数形式については触れない。固定小数点数形式では、小数点が置かれる位置を固定して数を表現する。

B.1.1 2 進数表現

数を表現するにあたって最も基本的な表現の一つが 2 進数表現であり、数を 0, 1 からなる有限個の文字（ビット）列として表現する。文字列の長さをビット長といい、ビット長 n で 2 進数表現できる数 x は、整数 m を用いて

$$x = \sum_{i=1}^{n} x_i 2^{n-i-m} \equiv x_1 x_2 \cdots x_{n-m}.x_{n-m-1} \cdots x_n \tag{B.1.1}$$

である。ただし $x_i \in \{0,\, 1\}$ である。これによって表現することのできる数の集合は $\{0,\, 1{\cdot}2^{-m},\, 2{\cdot}2^{-m}, \ldots, 2^{n-m} - 1\}$ である。本論文において小数は扱わないため、基本的に $m = 0$ とした非負の集合 $\{0,\ 1,\ 2, \ldots, 2^n - 1\}$ を表す際に用いる。

B.1.2 2 の補数表現

2 進数表現では負の数を扱えなかったが、2 の補数表現を用いることで負の数を表現することが可能になる。

$$x = -x_1 2^{n-1} + \sum_{i=2}^{n} x_i 2^{n-i} \tag{B.1.2}$$

は、ビット長 n における整数 x の 2 の補数表現を与えており、整数の集合 $\{-2^{n-1},\ -2^{n-1}+1, \ldots, 2^{n-1}-1\}$ を表すことが可能になる。本論文中で x が 0 以上か 0 未満かであるかを決定するビット x_1 を符号ビットと呼ばれる。表 3 ビットの時のビット列と整数の 2 進数表現、2 の補数表現との対応例を示す。

Bits	2進数表現における値	2の補数表現における値
000	0	0
001	1	1
010	2	2
011	3	3
100	4	−4
101	5	−3
110	6	−2
111	7	−1

表 B.1: 3ビットにおけるビット列と2進数表現、2の補数表現との対応

B.2　計算基底を用いた量子計算

古典計算においてビットに数を対応させることで計算を行うが、量子計算においてはビット以外にも数をエンコードする方法が存在する。例えば、確率振幅に数を導入する方法、回転ゲートの角度パラメータに数をエンコードする方法、状態の位相に数をエンコードする方法、さらにハミルトニアンへの符号化も存在する。それぞれ一長一短が存在するが、本論文では主に古典計算と同じく計算基底に数をエンコードする手法に基づいた計算を対象とする。この方法は、計算機科学分野においては量子計算をより弱めた形式である可逆計算としても今なお非常に活発な研究分野でもあり、効率的な基本的演算の構築でさえも様々な手法で考案されている。以下量子回路における基本的演算のための手法を説明する。

中でも第 2.3.2 節の Draper の加算回路をもとにし量子フーリエ変換をベースとした構成方法を説明する [68],[69]。この手法は、Draper の加算回路と同じく量子ビット数に関しては最良のものよりも、オーダーで1桁劣るがシステマティックに構成することができる利点があり、本文中での数値計算は主に以下の方法を用いている。ただし量子フーリエ変換をベースとした、古典アルゴリズムと本質的に異なる除算の構成方法は不明である。

B.2.1　加算・減算

第 2.3.2 節において Draper の加算回路で2つの非負整数に対する加算を量子フーリエ変換を用いて行う手法を紹介した。それを拡張し、ここでは任意の符号付きの整数に対する加算・減算を2の補数表現を用い構成する。

n_x, n_y $(n_x \geq n_y)$ ビットでそれぞれ表現された整数、計算基底 $|x\rangle, |y\rangle$ に対し

$$\mathrm{QNModAdd}_{n_x,n_y} |0\rangle |x\rangle |y\rangle = |x + y \bmod 2^{n_x+1}\rangle |y\rangle \tag{B.2.1}$$

$$\mathrm{QNModSub}_{n_x,n_y} |0\rangle |x\rangle |y\rangle = |x - y \bmod 2^{n_x+1}\rangle |y\rangle \tag{B.2.2}$$

を満たすユニタリ変換の構築を目指す。

そのために、CX ゲートを制御ビット $|x_1\rangle$、標的ビット $|0\rangle$ とすると、状態は $|x_1\rangle |x\rangle |y\rangle$ に変化する。$|x_1\rangle |x\rangle$ の部分にサイズ 2^{n_x+1} のフーリエ変換をすると、2π の位相の対称性に注意して

$$\sum_{k=1}^{2^{n_x+1}} \exp\left(2\pi \frac{1}{2^{n_x+1}} \left(x_1 2^{n_x} + x_1 2^{n_x-1} + \sum_{j=2}^{n_x} x_j 2^{n_x-j}\right)\right) |k\rangle = \sum_{k=1}^{2^{n_x+1}} \exp\left(2\pi \frac{x}{2^{n_x+1}} k\right) |k\rangle \tag{B.2.3}$$

となる。あとは位相 $\exp\left(\pm 2\pi \frac{y}{2^{n_x+1}} k\right)$ を加えればいい。そのために作用する量子ビットを $|k_i\rangle, |y_j\rangle$ とした制御位相

ゲートを $\mathrm{CP}_{i,j}$ として

$$V_{\pm} \equiv \bigotimes_{i=1}^{n_x+1} \bigotimes_{j=i}^{n_y} \mathrm{CP}_{i,j}\left(\pm(-1)^{\delta_{1j}}\frac{\pi}{2^{j-i}}\right) \tag{B.2.4}$$

を行えばよい。よって

$$\mathrm{QNModAdd}_{n_x,n_y} \equiv \left(\mathrm{IQFT}\otimes I^{\otimes n_y}\right) V_{+} \left(\mathrm{QFT}\otimes I^{\otimes n_y}\right) \left(CX_{1,0}\otimes I^{\otimes n_x+n_y-1}\right) \tag{B.2.5}$$

$$\mathrm{QNModSub}_{n_x,n_y} \equiv \left(\mathrm{IQFT}\otimes I^{\otimes n_y}\right) V_{-} \left(\mathrm{QFT}\otimes I^{\otimes n_y}\right) \left(CX_{1,0}\otimes I^{\otimes n_x+n_y-1}\right) \tag{B.2.6}$$

となる。

B.2.2 符号反転を伴う乗算

量子フーリエ変換を用いた乗算の構成方法を述べる。

n_x, n_y ビットでそれぞれ 2 の補数表現された二つの整数 x, y を考えると、$-2^{n_x+n_y-1} \leq -xy \leq 2^{n_x+n_y-1}-1$ が成り立つ。よって 2 の補数表現においては n_x+n_y-1 ビットで符号反転を含めた乗算 $\times(-1)\times$ が表現としては適している。$\times(-1)\times$ を量子回路内の計算基底で実現するために、n_x, n_y, n_x+n_y-1 ビットでそれぞれ表現された整数 x, y, z、計算基底 $|x\rangle, |y\rangle, |z\rangle$ に対し以下を満たすユニタリ変換 $\mathrm{QNModMul}_{n_x,n_y}$ として定義する。

$$\mathrm{QNModMul}_{n_x,n_y} |x\rangle |y\rangle |z\rangle = |x\rangle |y\rangle |z - xy \bmod 2^{n_x+n_y-1}\rangle \tag{B.2.7}$$

$N = 2^{n_x+n_y-1}$ とおく。計算基底 $|z\rangle$ に量子フーリエ変換を行うと計算基底 $|k\rangle$ に対し位相 $\exp\left(2\pi i z\frac{k}{N}\right)$ を持つ一様な重ね合わせ状態になる。(k はバイナリ表現として $0 \leq k \leq 2^{n_x+n_y-1}$ なる整数とする。) それぞれの基底に位相 $\exp\left(-2\pi i xy\frac{k}{N}\right)$ を加え、サイズ N の逆量子フーリエ変換を計算基底 $|k\rangle$ の重ね合わせ部分に作用させれば $\mathrm{QNModMul}_{n_x,n_y}$ と一致する。

位相 $\exp\left(-2\pi i xy\frac{k}{N}\right)$ に関し

$$\begin{aligned}
-xy\frac{k}{N} &= -\frac{1}{2^{n_x+n_y-1}}\left(-x_1 2^{n_x-1} + \sum_{i=2}^{n_x} x_i 2^{n_x-i}\right)\left(-y_1 2^{n_y-1} + \sum_{j=2}^{n_y} y_j 2^{n_y-j}\right)\sum_{l=1}^{n_x+n_y-1} k_l 2^{l-1} \\
&= -x_1 y_1 \sum_{l=1}^{n_x+n_y-1} k_l 2^{l-n_x-n_y} + x_1 \sum_{j=2}^{n_y}\sum_{l=1}^{n_x+n_y-1} y_j k_l 2^{-j+l} + y_1 \sum_{j=2}^{n_x}\sum_{l=1}^{n_x+n_y-1} x_j k_l 2^{-j+l} \\
&\quad - \sum_{i=2}^{n_x}\sum_{j=2}^{n_y}\sum_{l=1}^{n_x+n_y-1} x_i y_j k_l 2^{-i-j+l}
\end{aligned} \tag{B.2.8}$$

が成り立つ。四つの項は、それぞれ 0 か 1 の値をとる 3 つの数の積 $x_1y_1k_1, x_1y_jk_l, y_1x_jk_l, x_jy_jk_l$ に係数がかかったものから構成されている。つまり非ゼロであるためには 3 つの数全て非ゼロである必要がある。こうした位相を構成するには、多重制御位相ゲート

$$\mathrm{CCP}(\theta)|a\rangle|b\rangle|c\rangle = \begin{cases} \exp(i\theta)|a\rangle|b\rangle|c\rangle & (a=b=c=1) \\ |a\rangle|b\rangle|c\rangle & (\text{otherwise}) \end{cases} \tag{B.2.9}$$

を用いればいい。具体的には、作用する量子ビットが $|x_a\rangle, |y_b\rangle |k_c\rangle$ である多重制御位相ゲートを $\mathrm{CCP}_{a,b,c}$ として、$\mathrm{CCP}_{1,1,1}(-\pi)$ と、$2 \leq i \leq n_x, 2 \leq j \leq n_y, 1 \leq l \leq n_x+n_y-1$ を満たし加える位相が 2π の整数倍でないものに限定したゲート $\mathrm{CCP}_{1,j,l}\left(\pi 2^{1-j+l}\right)$, $\mathrm{CCP}_{i,1,l}\left(\pi 2^{1-i+l}\right)$, $\mathrm{CCP}_{i,j,l}\left(-\pi 2^{2-i-j+l}\right)$ で位相 $\exp\left(-2\pi i xy\frac{k}{N}\right)$ を加えることができる。

B.2.3　その他の初等演算

主に加減算回路を用いて構成できるものをここにまとめる。

n ビットの 2 の補数表現された x の符合反転 $x \to -x$ について。x の全ビットを反転した時の整数 $\bar{x}$ と関係式

$$\bar{x} = -2^{n-1}(1 - x_1) + \sum_{i=2}^{n-1} 2^{n-i}(1 - x_i) = -x - 1 \tag{B.2.10}$$

から、$\bar{x}$ に 1 を加算すれば $-x$ が構成できる。よって符合反転のためのユニタリ変換を

$$\mathrm{QSignInv}_n \equiv \left(I^{\otimes n+1} \otimes X\right) \mathrm{QNModSub}_{n+1,1} \left(I \otimes X^{\otimes n+1}\right) \tag{B.2.11}$$

$$\mathrm{QSignInv}_n \left|0\right\rangle \left|x\right\rangle \left|0\right\rangle = \left|-x\right\rangle \left|0\right\rangle \tag{B.2.12}$$

となる。(1 を加算することを -1 を減算することに置き換え補助レジスタのビット数を 2 ビットに抑えている。)

が成立するめ二つの数の大小比較については、減算を行い符合ビット n ビットの 2 の補数表現された x に対し絶対値 $|x|$ を求める演算について。

$$\mathrm{QAbs}_n \equiv C[\mathrm{QSignInv}_n]\,\mathrm{CX}_{2,n+1} \tag{B.2.13}$$

$$\mathrm{QAbs}_n \left|0\right\rangle \left|x\right\rangle \left|0\right\rangle = \left||x|\right\rangle \left|0\right\rangle \left|\text{sign bit of } x\right\rangle \tag{B.2.14}$$

ここで $C[\mathrm{QSignInv}_n]$ は制御ビットを $n+2$ 番目（右端）の量子ビットとし、標的ビットを残り $n+1$ 個の左端から順に全ての量子ビットとして標的ビットに条件付きで $\mathrm{QSignInv}_n$ を作用させることを指す。

大小比較のための回路について。整数 x, y とし $x - y$ の減算回路を構成し符合ビットを調べれば $x \geq y$ か $x < y$ が判定できる。$y - x$ の減算回路も構成し、$y \geq x$ か $y < x$ を調べ、$x > y, x = y, x < y$ を調べることができる。

その他の初等関数の構成として文献 [70] によれば、以下を満たす乗算のためのゲート U_{res}

$$U_{\mathrm{res}} \left|x\right\rangle \left|y\right\rangle \left|z\right\rangle = \left|x\right\rangle \left|y\right\rangle \left|z + xy\right\rangle \tag{B.2.15}$$

の存在を仮定した上で、x に関する関数 $1/x, \sqrt{x}, x^{\frac{1}{2^k}}, \ln x, x^c$ $(x \geq 1, k \in \mathbb{N}, c \in [0,1])$ が量子回路内で計算できる。これは古典計算と同じくニュートン法によるものである。例えば関数 $x \to 1/x$ を実装したければ、$f(w) = x - 1/w$ とすると $f'(w) = df/dw = 1/w^2$ より

$$w_{n+1} = w_n - \frac{f(w_n)}{f'(w_n)} = 2w_n - w_n^2 x \tag{B.2.16}$$

に従って w_n を更新していくことで $w_\infty = 1/x$ が得られる。ただニュートン法の反復回数に対し必要な量子ビットが線形に増大していくため、古典ビットと同じく可逆性を失わせて不必要なビットを捨てると重ね合わせ状態が崩れる欠点がある。

B.2.4　位相オラクルの構成方法

考案した量子勾配推定に必要な位相オラクルの構成方法を述べる。

関数値を位相に入力する式 (4.1.5) の位相オラクルを直接的に構成するには、位相ゲート P と（多重）制御位相ゲート CP があれば十分である。簡単のため特に関数が 1 変数かつ計算基底状態が n_1 ビットの 2 進数表現で表され

ている場合を以下に示す。

$$O_{2\pi Sf} \equiv \prod_{j\geq 0} O_{2\pi Sf_j} \tag{B.2.17}$$

$$f_j(x) \equiv a_j x^j \tag{B.2.18}$$

$$O_{2\pi Sf_0} \approx I \tag{B.2.19}$$

$$O_{2\pi Sf_1} \equiv \prod_{k} P_k(2\pi S a_1 2^{n_1-k}) \tag{B.2.20}$$

$$O_{2\pi Sf_2} \equiv \prod_{k,l} \mathrm{CP}_{k,\,l}(2\pi S a_2 2^{2n_1-k-l}) \tag{B.2.21}$$

ここで定数関数 (B.2.19) は大域的位相因子を除き一致するとしている。3 次以上の多項式関数の場合も同様である。（制御）位相ゲートに対する下付き添字は計算基底状態 $|x\rangle$ の対応する量子ビットに作用することを意味する。

さらに位相オラクルは以下で定義するバイナリーオラクル

$$O_{\text{binary-}f} |\boldsymbol{x}\rangle |\boldsymbol{0}\rangle = |\boldsymbol{x}\rangle |f(\boldsymbol{x})\rangle \tag{B.2.22}$$

の 2 回の使用と式 (B.2.22) 内の n_2 ビットの $|f(\boldsymbol{x})\rangle$ の各量子ビットに作用する位相ゲート P

$$O^{-1}_{\text{binary-}f} \prod_j P_j\left(2\pi S 2^{n_2-j}\right) O_{\text{binary-}f} |\boldsymbol{x}\rangle |\boldsymbol{0}\rangle = O_{2\pi Sf} |\boldsymbol{x}\rangle |\boldsymbol{0}\rangle \tag{B.2.23}$$

により間接的に構成可能である。そのため要請した位相オラクルは直接・間接を問わず実装は容易である。

付録 C

最適化問題

最適化問題とは特定の集合上である実数値関数（“目的関数” と呼ばれる）が最小もしくは最大となる解を求める問題である。（大域的最適化に関して文献 [71, 72] を参考にした。）

定義 C.0.1. 一般に最適化問題とは、集合 X とその空でない部分集合 $S \subseteq X$ として関数 $f\colon S \to \mathbb{R}$ に対して

$$\min_{x \in S} f(x) \ (or \ \max_{x \in S} f(x)) \tag{C.0.1}$$

を行う問題を最適化問題（*optimization problems*）と呼ぶ。

$f(x)$ は目的関数と呼ばれ、要素 x に対するある定量的尺度である。また集合要素が部分集合上 $S \subseteq X$ でとることを課されているため、$S \subseteq X$ を制約条件（constraint）と呼ばれ、S を可能領域 (feasible region) というが、本文では領域と略す。可能領域に含まれる x を実行可能解 (feasible solution) といい、f を最小または最大化する実行可能解 x を最適解 (optimal solution) という。

最適化問題は一般に大きく離散的であるか、連続的であるかに分類され、その上で目的関数が線形であるか、非線形であるかに分かれる。連続関数である場合には問題を連続最適化問題とよび、凸（conjective）か非凸かに分類し非凸な問題は大域的最適化問題（global optimization）とされる。一方離散集合に対する最適化問題を組合せ最適化問題という。離散集合上の関数に対しては凸性に対応する概念が存在しないが、離散集合上での各要素の近傍の要素に対して凸性の概念を拡張することで同じく大域的最適化問題を定義することができる。本論文では、離散かつ有限な集合 S のみを扱うとし、実数の集合 $\mathbb{R}$ についてはグリッドに分割し有限個の要素を取り出した集合として考える。以下にその定義を記す。

定義 C.0.2. S を空でない有限集合として関数 $f\colon S \to \mathbb{R}$ に対して

$$\min_{x \in S} f(x) \tag{C.0.2}$$

を行う問題を “（有限）大域的最適化問題（*finite global optimization problems*）” と呼ぶ。

本論文ではこの有限集合上で定義された大域的最適化問題を考えることとする。大域的最適化問題における最適解を大域的最適解という。局所最適解を定義するためには、最適化問題における近傍の定義方法と解法に依存する。そのためその都度定義することとしてここでは明確に定めない。

本論文では上記の最適化問題に限って議論を行う。

最適化における最も重要な事実としてノーフリーランチ定理 [73],[74] が挙げられる。それによれば、どの最適化手法でも平均的には性能に優劣がないという定理である。それは量子計算においても同様だと考えられる [75]。しかし多くの現実的問題において可能領域に何らかの構造（例えば順序構造）が存在することが多いため、特定の問題に特

化した最適化手法を構築することが重要である。

大域的最適化問題において最も基本的な手法が力任せ (brute-force) またはしらみつぶし (exhaustive) 探索 であり、可能領域内の解を全て調べ上げる方法である。しかし最適化問題の例として巡回セールスマン問題の場合は計算量が階乗関数としてスケールするため現実的に不可能である。さらに連続変数に対する大域的最適化問題において調べ上げることは不可能なため、領域を格子状に離散化し格子点上で調べ上げる手法が存在し grid search という。グリッドが高次元の n 次元の超直方体をなすとすれば、指数的に調べ上げる操作を要するため問題点とされている。

確率的な grid search として知られるのが、pure random search(PAS) であり、1958 年に Brooks[76] により初めて提案された最も単純な確率的アルゴリズムである。pure random search は領域から繰り返しサンプルをとり探索を行うもので、典型的には一様ランダムに選択されるものを指す。

pure random search は各サンプリングは独立な試行であったが、それまでのサンプリング結果に依存する探索手法として pure adaptive Search が存在する。**Algorithm 8** にそれを示す。

Algorithm 8 Pure Adaptive Search, **PAS**

$x_1 \in S$ を S 上で一様ランダムに選択し、$y_1 = f(x_1)$ とする
for $i = 1, 2, \ldots$ *until a termination condition is met* **do**
　集合 $S_i = \{x \in S \colon f(x) < y_i\}$ から一様ランダムに x_{i+1} を取りだし、$y_{i+1} = f(x_{i+1})$ とおく。
end for

局所最適化に対し非常に多くの手法が存在するため、ここではその一部を紹介する。まず局所的探索による手法として山登り法、焼きなまし法 (simulated annealing)、タブーサーチが存在する。それぞれ簡単に説明すると、山登り法では近傍の中で最も最適な解を選択し逐次更新する方法である。焼きなまし法は、現在の解の近傍からランダムに選択しより良い解が得られれば解を更新し、そうでなくても “温度” にパラメトライズされた遷移確率に従って近傍内の解へ更新する方法である。その際更新が進むにつれて “温度” を低下させ遷移を起こりにくくすることで、局所解に陥る効果を抑えることができる。これに関連して “温度” の代替として量子揺らぎを利用した量子アニーリング [77],[78] という手法も存在する。タブーサーチとは、現在の解から仮に改善されなくともその近傍の中で最良の解へ更新する。局所最適解の近傍では、複数の更新後過去に探索した解へ戻ってきてしまう現象 “サイクリング” が起こってしまう。それを避けるためタブーサーチでは、タブーリストと呼ばれる解の集合の定義しタブーリストに含まれる解への遷移を禁止することで、サイクリングを避ける。またその他の局所探索手法として形式的には属さないとされることの多いが、遺伝的アルゴリズムという手法も存在する。それは、生物の自然淘汰、交叉、突然変異を模倣したもので遺伝子に相当する評価関数に基づき解の更新を行う方法である。いずれも解を近傍へ更新することで最適化をするため局所解への停留問題を何らかの方法で避けている。

局所最適化において関数の勾配情報を用いることができる場合は、勾配法といった手法が存在する。勾配法においてもさまざまな工夫によって局所解に陥りやすい難点を克服するための工夫が非常に多く存在する。

それでもなお局所最適化手法は局所的最小値に陥りやすいという欠点があるが、それを克服するために初期解をランダムに複数回サンプリングする手法があり、multi-start method と呼ばれる [79] (**Algorithm 9** 参照)。手法自体はよく知られたものであるがその名はあまり知られていない。本論文中では、最終的に multi-start 法における量子加速について議論する。

Algorithm 9 Multi-start method による最小化

$f_* \leftarrow \infty$
for $i = 1, 2, \ldots$ *until a termination condition is met* **do**
  $x_i \in S$ を S 上で一様ランダムに選択する。
  x_i から局所最適化をスタートし、局所最適解 x を得る。
  if $f(x) < f_*$ **then**
    $x_* \leftarrow x, f_* \leftarrow f(x)$
  end if
end for