

量子回路マッピングの最適化

Minimum Cost Mapping of Quantum Circuits to Quantum Computers

井床 利生^{*1}
Toshinari Itokoルディ レイモンド^{*1}
Rudy Raymond今道 貴司^{*1}
Takashi Imamichi^{*1}IBM 東京基礎研究所
IBM Research - Tokyo

Recent technological progress has brought us closer to quantum computers that can outperform current classical computers. However, such quantum computers in the near future will likely be very noisy and suffer from limited coupling architecture. A quantum algorithm must be compiled into a quantum circuit, most often by adding supplementary quantum gates, to satisfy the coupling restriction of quantum hardware. We propose a novel formulation of mapping quantum circuits to quantum computers by introducing a dependency graph that represents the partial order structure of the circuit essential in the mapping. The formulation enables us to leverage techniques in combinatorial search to find the mapping with minimum cost of additional gates. We obtain better solutions for cost-optimal mapping of standard benchmark circuits.

1. はじめに

近年の量子コンピューターのハードウェア技術の進展により、ノイズ付きで量子ビット数が中小規模の量子コンピューター（近似量子コンピューター）が実現し、量子アルゴリズムを実機で動かせる時代が到来した。誤り耐性を備えた理想的な万能量子コンピューターの実現にはまだ相当の年月を要すると考えられているが、数年内に実現される近似量子コンピューターの精度や規模でも、量子化学やサンプリングなど特定の分野のタスクでは古典コンピューターを上回る計算能力を持つのではないかと期待されている。近似量子コンピューターについては [Preskill 18] を参照されたい。

一般に量子アルゴリズムを実装した量子プログラムを量子コンピューター上で実行する場合、量子回路にコンパイルする必要がある。量子回路は、図 1a のようなダイアグラムで表現される。量子ビットに対応する水平線を描き、各量子ビットに左から右へと作用させる順にゲートを配置する。図 1a の例では、2 量子ビットゲートの CNOT（制御ビットを ●、ターゲットビットを ⊕ と表記）と 1 量子ビットゲートのアダマールゲート（ H と表記）が使われている。本研究では、量子ゲートとして、1 量子ビットゲートおよび CNOT のみを考える。これは任意の量子計算をこれらのゲートのみで構築可能であるという事実に基づく。なお、本研究では量子回路の古典的な組合せ構造にのみ着目するため、量子回路で実現される量子計算の内容には立ち入らない。量子計算の詳細については例えば [Nielsen 10] を参照されたい。

量子プログラムのコンパイル時には、量子コンピュータの物理的制約を満たすために、与えられた量子回路を物理的に実行可能な等価な量子回路に変換する必要がある。そのような変換を量子回路マッピングと呼ぶ。主要な物理的制約として、物理的に隣接する量子ビット間にしか CNOT を配置できないという接続制約がある。接続制約は、図 1b のような、量子ビットを節点とし CNOT を実行可能な量子ビット対を枝とした有向グラフ（接続制約グラフ）として表現される。枝の始点が制御ビット、終点がターゲットビットに相当する。例えば、図 1a の回路上の量子ビット b_{1-4} を図 1b の物理的量子ビット

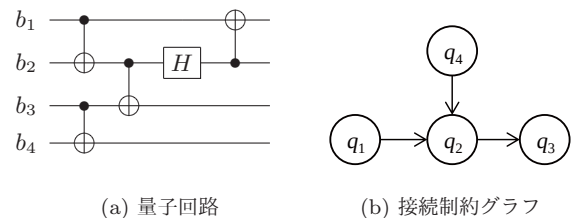


図 1: 量子回路 (a)、および、接続制約グラフ (b)

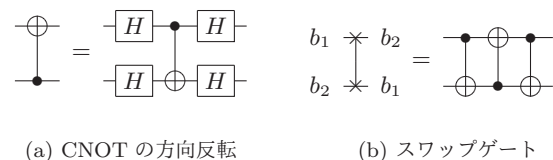


図 2: 接続制約の充足手段

q_{1-4} に配置した場合、 b_3 から b_4 への CNOT と b_2 から b_1 への CNOT はそのままでは実行できない。後者は図 2a のように CNOT の前後にアダマールゲートを挿入し方向を逆転させることで実行可能にできる。前者を実行可能にするには、接続制約グラフ上で b_3 と b_4 配置が隣接するまで量子ビット対のスワップを行う必要がある。そのようなスワップは図 2b に示すスワップゲートによって実現できる。いずれにせよ接続制約を満たすマッピングを行うには、元の回路にゲートを追加する必要がある。しかし、ゲートの追加はノイズおよび回路長の増大に直結する（特に誤り耐性を持たない近似量子コンピューターにおいてノイズの増大は計算結果の誤り率の増加を意味する）ため最小限に抑えることが求められる。その意味で量子回路マッピングにおける追加ゲート数の最小化は量子コンピューターの実効性能に影響を与える重要な技術といえる。

本研究では、現在の近似量子コンピューターにおいて、1 量子ビットゲートに比べ CNOT の誤り率が数倍～数十倍も大きいことを踏まえ、スワップゲートの追加コストのみに注目し、

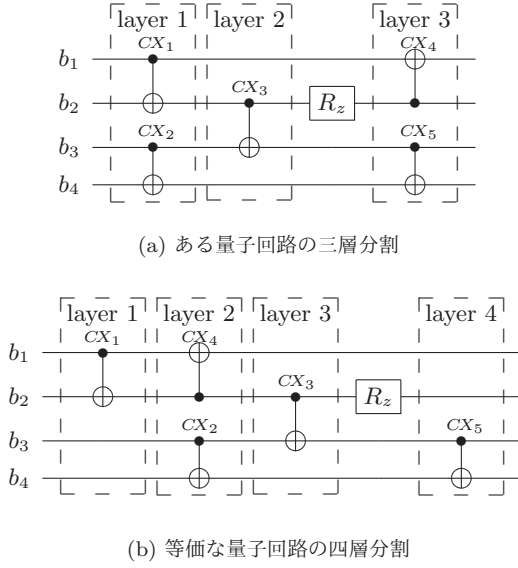


図 3: 等価な量子回路に対する異なる層分割

接続制約の下で最小の追加スワップゲート数（最小スワップ数）を達成する量子回路マッピングを求める問題を扱う。そして当該問題に対して、ゲートの実行順序の可換性を考慮した新しい定式化を与えると共に、その定式化に基づく厳密解法を提案する。

2. 背景

量子回路マッピングに関する先行研究の多くは、接続制約グラフの構造を一次元パスや二次元格子に限定したものが多かったが、2017 年に公開された IBM Q systems [IBM Q 17] の接続制約グラフがそのような特殊な構造を持たなかったこともあり、グラフ構造を限定しない問題に対する研究が 2017 年以降相次いで発表された [Bhattacharjee 17, Zulehner 17, Siraichi 18]。しかし、それらの研究はいずれもゲートの実行順序の可換性を考慮しておらず、また、入力回路に対する層分割を固定した定式化に基づいており、最適値が層分割の与え方に依存するという課題があった。ここで量子回路の層 (layer) とは (接続制約がない場合に) 並列実行可能な CNOT の集合のことである。

図 3 に等価な二つの回路に対して異なる層分割を行った場合に、異なる最小スワップ数が得られる例を示す (R_z は 1 量子ビットゲートの一種)。図 3a の回路は三層、図 3b は四層だが、図 4 に示す 2 量子ビットゲートの可換則を考慮する (CX_4 に対して図 4a、図 4b の可換則を順に適用する) と等価であることが分かる。そして、図 3a の層分割の下での最小スワップ数は 2 であるが、図 3b の層分割の下での最小スワップ数は 1 である。

我々は上記課題を克服する定式化を次節で提案する。

3. 定式化

まず定式化に必要な用語を定義する。図 1a のように量子回路マッピングにおいて入力として与えられる回路を入力量子回路と呼び、入力量子回路上の量子ビット集合 (入力量子ビット) を $B = \{b_1, \dots, b_{|B|}\}$ で表す。また、出力として得られる物理的に実行可能な量子回路を物理量子回路と呼び、物理量子回路上の量子ビット集合 (物理量子ビット) を $Q = \{q_1, \dots, q_{|Q|}\}$

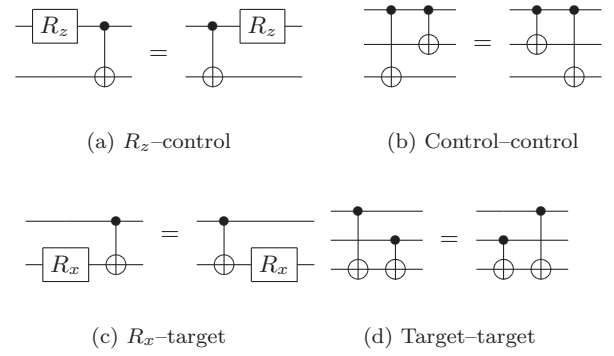


図 4: 量子ゲートの代表的な可換則

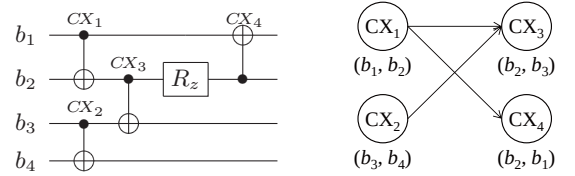


図 5: 入力量子回路 (左) とその依存グラフ (右)

で表す。入力量子ビット B の物理量子ビット Q への割り当てを配置 (layout) と呼び、 $(b_1, b_2, b_3) \mapsto (q_1, q_2, q_3)$ 、もしくは入力量子ビットの順序を固定して単純に (q_1, q_2, q_3) と表す。因みに、入力量子ビット数と物理量子ビット数は同じと仮定できる ($|B| < |Q|$ の場合は B に補助ビットを追加し $|B| = |Q|$ とできる) ので、配置は順列ともみなせる。図 1b のように物理量子ビットを頂点とし CNOT を実行可能な対を枝としたグラフを接続制約グラフと呼び、 $C = (V_C, E_C)$ で表す。ここで C の頂点集合 $V_C = Q$ である。なお、本研究では CNOT の方向反転に要するコストは考慮しないため、以降では接続制約グラフを無向グラフとみなす。

一般的に量子回路マッピングは、与えられた入力量子回路に対し、接続制約グラフで表される制約を満たすような物理量子回路への変換を求める問題として定式化される。我々は、前節で見た課題を解決する定式化を行うため、入力量子回路と図 4 に示す量子ゲート可換則から求められる依存グラフ $D = (V_D, E_D)$ を定義する。図 5 に示すように、依存グラフ D の頂点集合 V_D は CNOT の集合であり、各 CNOT には作用対象の量子ビット対がラベルとして付与される。また、枝集合 E_D には CNOT の実行順序に対する依存関係、すなわち、 $(CX_i, CX_j) \in E_D$ ならば CX_i を必ず CX_j より先に実行する必要がある、という関係が保持される。なお、[Matsuo 11] で上記依存グラフに似たグラフが考えられているが、量子ゲート可換則が考慮されていない点で明確に異なっている。

依存グラフを用いることで、最小スワップ数の量子回路マッピングを求める問題は、次のような組合せ最適化問題として定式化できる。

最小スワップ数量子回路マッピング 接続制約グラフ C および入力量子回路の依存グラフ D が与えられたもとで、 C を充足するような、初期配置 l_0 およびゲート系列 I_D で、 I_D 中のスワップゲート数が最小となるものを求めよ。

ここで、 D に対するゲート系列 I_D とは、CNOT およびスワップゲートのリストで、全ての CNOT の順序が D の与える半

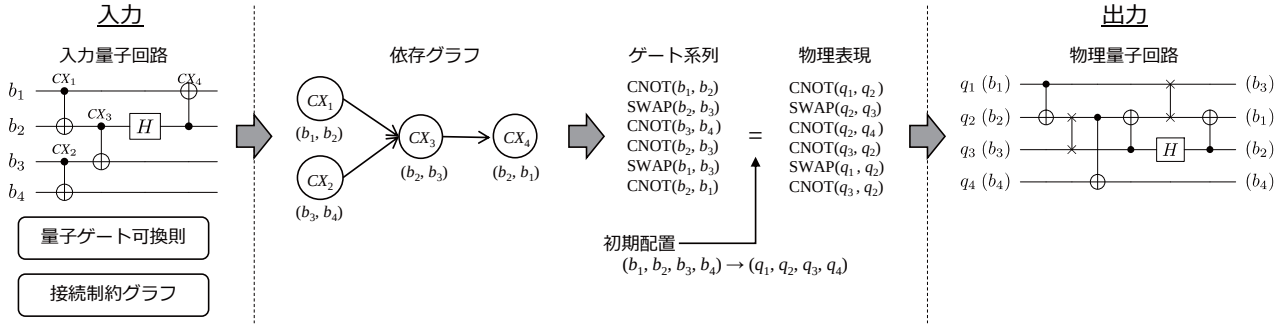


図 6: 量子回路マッピングの流れ

順序関係に矛盾しない（つまりトポロジカル順序をなす）ものとする。また、ゲート系列 I_D の全てのゲート系列の作用量子ビットは入力量子ビット B で表現されるが、初期配置 l_0 を与えることで、作用量子ビットを物理量子ビット Q を用いた表現に変換することもできる。そのような表現を初期配置 l_0 に対する物理表現 $I_D^{ph}(l_0)$ と呼ぶことにする。初期配置 l_0 およびゲート系列 I_D が C を充足するとは、 $I_D^{ph}(l_0)$ の全てのゲートの作用量子ビット対が C の枝になっていることを意味する。また、依存グラフの構築の際に 1 量子ビットゲートの情報を記憶しておけば、得られた $I_D^{ph}(l_0)$ から最終出力である物理量子回路を復元することは難しくない。

以上、量子回路マッピングの全体の流れを図 6 にまとめる。

4. アルゴリズム

本節では、最小スワップ数量子回路マッピングの厳密解法を提案する。まず、状態 (l, R) を、配置 l と残存 CNOT 集合 R のペアとして定義する。但し、残存 CNOT 集合は、依存グラフ D を半順序集合と見た時の上方集合 (upper set) に限定した V_D の部分集合とする。例えば、図 6 中の依存グラフの残存 CNOT 集合は、次のいずれかである： $\{CX_1, CX_2, CX_3, CX_4\}$ 、 $\{CX_2, CX_3, CX_4\}$ 、 $\{CX_1, CX_3, CX_4\}$ 、 $\{CX_3, CX_4\}$ 、 $\{CX_3\}$ 、 $\{CX_4\}$ 、 $\emptyset$ 。また、残存 CNOT 集合 R と配置 l に対して、 R からの l に対する残存 CNOT 集合を、配置 l の下で接続グラフ C 上で実行可能な先頭の CNOT を R から除いた残りの CNOT 集合と定義する。例えば、図 6 の依存グラフにおいて、 $\{CX_1, CX_2, CX_3, CX_4\}$ からの配置 (q_1, q_2, q_3, q_4) に対する残存 CNOT 集合は、先頭の CX_1 を実行できるが他は実行できないため $\{CX_2, CX_3, CX_4\}$ となる。状態間の遷移はスワップゲートの挿入によって発生する。例えば、状態 $((q_1, q_2, q_3, q_4), \{CX_2, CX_3, CX_4\})$ から、 b_2 と b_3 をスワップし q_3 と q_2 に配置を変え、状態は $((q_1, q_3, q_2, q_4), \{CX_4\})$ に遷移する。

任意の初期状態から出発し状態 (l, R) に到達するのに必要な最小スワップ数を $f(l, R)$ で表すとする。ここで初期状態は、任意の配置 l と、 V_D からの l に対する残存 CNOT 集合 R_0 を用いて、 (l, R_0) と表される状態とする。すると、最小スワップ数量子回路マッピングの最適値は、あらゆる配置 l に対する $f(l, \emptyset)$ の最小値となる。その値はアルゴリズム 1 で求められる。アルゴリズムはまず初期状態に対する f の値を 0 に設定する。次に一回のスワップで到達可能な状態の f の値を更新し、その次に二回のスワップで到達可能な状態の f の値を更新し、と順に続ける。空の残存 CNOT 集合を持つ状態に到達したら終了する。最適値だけでなく最適解すなわちゲート系列 I_D を得る必要がある場合には、17 行目の直前で $f(l', R') > 1 + f(l, R)$

アルゴリズム 1 最小スワップ数量子回路マッピングの厳密解法

IN: 量子ビット B をもつ入力量子回路の依存グラフ $D = (V_D, E_D)$ および接続制約グラフ $C = (Q, E_C)$

OUT: C を充足する D のマッピングにおける最小スワップ数

```

1:  $S \leftarrow \emptyset$  // アクティブ状態
2: for all  $B$  の  $Q$  への配置  $l$  do
3:    $R_0 \leftarrow V_D$  からの  $l$  に対する残存 CNOT 集合
4:    $f(l, R_0) \leftarrow 0$ 
5:    $S \leftarrow S \cup \{(l, R_0)\}$ 
6: end for
7: while True do
8:    $S' \leftarrow \emptyset$ 
9:   for all 状態  $(l, R) \in S$  do
10:    for all 枝  $e \in E_C$  do
11:       $l' \leftarrow$  配置  $l$  から  $e$  をスワップした後の配置
12:       $R' \leftarrow R$  からの  $l'$  に対する残存 CNOT 集合
13:      if  $f(l', R')$  が未定義 then
14:         $f(l', R') \leftarrow +\infty$ 
15:         $S' \leftarrow S' \cup \{(l', R')\}$ 
16:      end if
17:       $f(l', R') \leftarrow \min\{f(l', R'), 1 + f(l, R)\}$ 
18:      if  $R' = \emptyset$  then
19:        return  $f(l', R')$ 
20:      end if
21:    end for
22:  end for
23:   $S \leftarrow S'$ 
24: end while

```

の場合に、直前の状態 (l, R) から現在の状態 (l', R') を関連付けることで、 e (スワップゲート) と $R' \setminus R$ (CNOT 集合) を記憶しておけばよい。

5. 実験

前節に示したアルゴリズムの性能および前々節に示した定式化の有効性を確認するための実験を行った。アルゴリズムは量子計算向けソフトウェア開発キットである QISKit [QISKit 17] をベースに Python 言語で実装した。実験はノート PC (Intel Core i5-5300U CPU 2.3 GHz, 8 GB メモリ) 上で行った。実験に用いた全てのインスタンスに対して、各計算に要した時間は 30 秒以内であった。

表 1 に既存手法 [Shafaei 14, Lye 15] および提案手法を用いて二次元格子接続制約の下で量子回路マッピングを求めた場合

の必要スワップ数を示す。量子回路データセットとしては、比較対象の先行研究で用いられている RevLib [Soeken 12] を採用し、QISKit でサポートされる OpenQASM 形式 [Cross 17] に変換した上で使用した。全回路に対して混合整数計画 (MIP) ベースのヒューリスティクスである [Shafaei 14] と同等かよりよい結果が得られることを確認した。回路 3_17_13 と hwb4_52 では、Pseudo-Boolean Optimization (PBO) を用いた厳密解法である [Lye 15] よりもよい結果が得られた。原因としては、前処理の回路の変換において、3 量子ビット以上のゲートを 2 量子ビット以下のゲートに分解する規則が一意でないことが考えられるが、[Lye 15] に分解規則についての記述がないため断定はできない。

表 1: 二次元格子接続制約下での量子回路マッピングの解法毎のスワップ数

回路	B	格子形状	MIP ^a	PBO ^b	提案手法
3_17_13	3	2×2	6	4	3
4_49_17	4	2×2	13	—	7
4gt11_84	5	2×3	2	1	1
4mod5-v1_23	5	2×3	11	—	7
decod24-v3_46	4	2×2	3	2	2
hwb4_52	4	2×2	9	7	6
rd32-v0_67	4	2×3	2	2	2

^a MIP [Shafaei 14]. ^b PBO [Lye 15].

加えて、本論文で提案した定式化によって層を固定した定式化よりも最小スワップ数が減少することを確認するために、人工ランダム回路を用いた実験を行った (表 2)。データとしては、 R_z 、 H 、CNOT をそれぞれ 25%、25%、50% の確率で 100 個生成したランダム回路 10 セットを用いた。表中の最小スワップ数は 10 セットでの平均値である。アルゴリズムは、両定式化に対してそれぞれの厳密解法を実装して用いた。接続構造によらず本論文で提案した定式化によって、よりよい量子回路マッピングが行えることを確認できた。

表 2: 最小スワップ数の定式化による違い

B	接続構造	層固定定式化	提案定式化
6	一次元パス	32.4	29.6
5	一次元パス	25.9	23.5
6	二次元格子 (2×3)	12.6	11.9
5	ibmqx4 [IBM QX4 17]	10.4	8.0

6. おわりに

本研究では、量子ゲートの可換則を考慮した最小スワップ数量子回路マッピングの新しい定式化とその定式化に基づく厳密解法を提案し、計算機実験によりその有効性を確認した。今後の研究課題としては以下が考えられる。定式化に関しては、目的関数の拡張 (例えば、1 量子ビットゲートのコストや回路の層数を考慮する) や、考慮する可換則の拡張 (より複雑な三つ以上の量子ゲートが関係するような可換則も考慮する) などが考えられる。アルゴリズムに関しては、今後のハードウェアの量子ビット数の増大に対応できるような性能のよいヒューリスティクスの開発が重要になってくると考えられる。

参考文献

- [Bhattacharjee 17] Bhattacharjee, D. and Chattopadhyay, A.: Depth-Optimal Quantum Circuit Placement for Arbitrary Topologies, *arXiv preprint arXiv:1703.08540* (2017)
- [Cross 17] Cross, A. W., Bishop, L. S., Smolin, J. A., and Gambetta, J. M.: Open Quantum Assembly Language, *arXiv preprint arXiv:1707.03429* (2017)
- [IBM Q 17] IBM Q, : IBM Q (2017), <https://www.research.ibm.com/ibm-q/>
- [IBM QX4 17] IBM QX4, : IBM QX4: Raven (2017), <https://github.com/QISKit/ibmqx-backend-information/blob/master/backends/ibmqx4/README.md>
- [Lye 15] Lye, A., Wille, R., and Drechsler, R.: Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits, in *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pp. 178–183 (2015)
- [Matsuo 11] Matsuo, A. and Yamashita, S.: Changing the gate order for optimal LNN conversion, in *International Workshop on Reversible Computation*, pp. 89–101 (2011)
- [Nielsen 10] Nielsen, M. A. and Chuang, I. L.: *Quantum Computation and Quantum Information*, Cambridge University Press (2010)
- [Preskill 18] Preskill, J.: Quantum Computing in the NISQ era and beyond, *arXiv preprint arXiv:1801.00862* (2018)
- [QISKit 17] QISKit, : Quantum Information Software Kit (2017), <https://www.qiskit.org/>
- [Shafaei 14] Shafaei, A., Saeedi, M., and Pedram, M.: Qubit placement to minimize communication overhead in 2D quantum architectures, in *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, pp. 495–500 (2014)
- [Siraichi 18] Siraichi, M., Dos Santos, V. F., Collange, S., and Pereira, F. M. Q.: Qubit Allocation, in *CGO 2018-IEEE/ACM International Symposium on Code Generation and Optimization*, pp. 1–12 (2018)
- [Soeken 12] Soeken, M., Frehse, S., Wille, R., and Drechsler, R.: RevKit: An Open Source Toolkit for the Design of Reversible Circuits, in *Reversible Computation 2011*, Vol. 7165 of *Lecture Notes in Computer Science*, pp. 64–76 (2012), RevKit is available at www.revkit.org
- [Zulehner 17] Zulehner, A., Paler, A., and Wille, R.: Efficient Mapping of Quantum Circuits to the IBM QX Architectures, *arXiv preprint arXiv:1712.04722* (2017)