

# デジタルアニーラを用いた量子状態測定の高速度の取り組み

栗田知周<sup>1</sup> 森田幹雄<sup>1</sup> 大島弘敬<sup>1</sup> 佐藤信太郎<sup>1</sup>

**概要:** 量子コンピュータの計算規模を拡大したときの時の問題点の一つとして、膨大な数の observable 測定が必要になり、それに比例して計算時間が長くなる点がある。近年、observable を同時測定が可能な組に分割 (partitioning) することで、量子状態測定を高速度化する手法が考案された。しかし、その分割は NP 困難な問題として知られており、observable 数が増えるとその partitioning 問題を解くのに必要な時間が膨大になるという問題があった。本発表では、量子状態測定の高速度を目指し、デジタルアニーラを利用して、partitioning を高速に行う手法を提案する。

**キーワード:** 量子コンピューティング, 同時測定, partitioning, デジタルアニーラ

## Fast Measurement of Quantum States Using Digital Annealer

TOMOCHIKA KURITA<sup>1</sup> MIKIO MORITA<sup>1</sup> HIROTAKE OSHIMA<sup>1</sup>  
SHINTARO SATO<sup>1</sup>

**Abstract:** One of the problems when the calculation scale of a quantum computer becomes large is that a huge number of observable measurements are required, and the calculation time increases in proportion to this. Recently, simultaneous measurement techniques for fast quantum-state measurement have been proposed. It is known, however, that the problem of making a minimum number of groups of observables that can be simultaneously measured, which is called “partitioning”, is NP-Hard. In this report, we propose a method for partitioning with Digital Annealer, which enables this step to proceed fast.

**Keywords:** Quantum Computing, Simultaneous Measurement, Partitioning, Digital Annealer

### 1. はじめに

量子コンピュータは量子化学計算をはじめとする分野で古典コンピュータの性能を凌駕する可能性を秘めており、そのための新たなアルゴリズムの開発が進められている。

量子コンピュータでの計算速度を律速しているものの一つに測定操作がある。量子状態の直接測定については、以下の2つの制約がある：(i) 測定は computational basis のみしか行えない。computational basis では観測できない成分を測定したい時は、観測したい成分が computational basis で測定できるように基底変換の演算をしてから測定をする必要がある。(ii) 状態測定により量子状態は破壊されてしまう。そのため、演算後の量子状態について測定したい成分が複数ある場合、必要な全測定回数 (ショット数) はその成分の数に比例する。例えば、ある  $n$  量子ビットの状態に対して full tomography を行う場合、測定対象の成分の数は  $4^n - 1$  となり、よって必要なショット数は量子ビット数に対し指数関数的に増大する。

このような測定対象成分の増大による必要ショット数の増大を抑制するため、一つの量子回路で複数の成分を同時に測定する手法が提案された[1][2]。複数の成分に対応するパウリ演算子列がそれぞれ可換であれば、それらは同時測定が可能となる。この同時測定手法を利用した observable 測定の流れを図1に示す。図1におけるステップ③に示す

ように、測定対象成分に対応する observable 群を、下記(A)を満たす部分群 (clique) に分割する (partitioning) が必要になる。

「clique 内の任意の2つの observable につき、その2つは可換になる」…(A)

この際、clique の数が少ないほど、ステップ④⑤の同時測定を効果的に行うことができる。一方で、この clique 数を最小にするように分割する問題は NP 困難な問題とされている[1][2]。これまで、このような partitioning 手法として、Boppana, Halldórsson らにより提案された手法[3]や Bron, Kerbosch らにより提案された手法[4][5]が知られている。しかし、それらが本問題を解くのに必要な時間の observable 数に対するスケールは、それぞれ多項式時間のおよび指数関数的になる。よって、数千量子ビットを超えると、その計算に一般的なコンピュータでは数時間以上かかってしまい、双方とも量子コンピュータの大規模化した際の同時測定には向いていない。

また、量子化学計算 (VQE) 用に特化した分割方法も提唱されている[2]。しかしながら、今後、新規量子アルゴリズムの研究開発が進むにつれて量子コンピュータの活用領域が広がると期待されている点から、特定のアルゴリズムによらない、汎用性のある partitioning 手法が必要になると考えられる。

<sup>1</sup> 富士通株式会社、研究本部、量子コンピューティング研究センター  
Quantum Computing Research Center, Fujitsu Research, Fujitsu Limited

本研究報告で我々は、この分割問題をより高速、効果的に解くため、問題をイジングモデルに落とし込み、量子インスパイアコンピュータであるデジタルアニーラ (DA) [6] を活用する方法を提案する。DA はイジングモデルにマッピングされた組合せ最適化問題を高速に解くための専用アーキテクチャであり、問題をうまく落とし込めれば、既存の手法と比較して大幅な高速化が期待できる。本研究報告では、DA を用いた partitioning を行い、その結果および所要時間について、Boppana-Halldórsson アルゴリズム[3]や Bron-Kerbosch アルゴリズム[4][5]と比較し、考察を行った。

なお、observable 間の可換性については、Qubitwise Commute (QWC)、General Commute (GC) の 2 通りの定義がある[2]。本報告では、GC の定義で partitioning を行った方がより clique 数が少なくなるので、GC の定義に沿った partitioning を検討した。

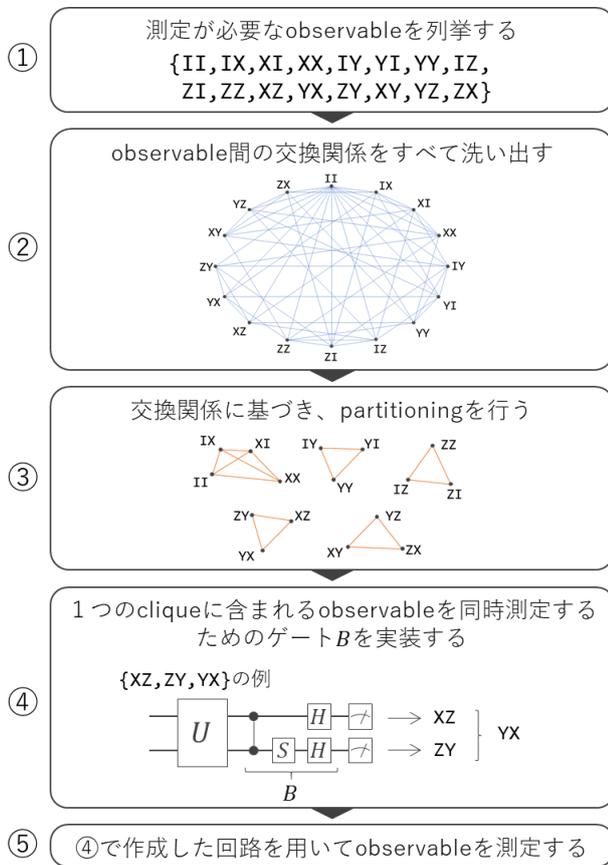


図 1. 同時測定手法を用いた observable 測定の流れ

## 2. 方法と理論

本章では、本研究報告をするにあたり必要となる技術的なバックグラウンドを述べる。まず、2.1 節では、同時測定が可能であるためには observable 同士が可換であることが必要な理由を述べる。次に、2.2 節では、partitioning において本研究報告で提案する手法および従来手法について、その概要を述べる。そして、2.3 節では、本報告で対象とした

observable 群を示す。

### 2.1 同時測定の理論

一般に、量子状態  $\rho$  の observable  $M$  による期待値は、

$$\text{Tr}(M\rho)$$

となる。ここで、 $M = B^{-1}ZB$  となるような  $B$  を導出すると、

$$\text{Tr}(M\rho) = \text{Tr}[Z(B\rho B^{-1})] \quad \dots (1)$$

となる。すなわち、 $\rho$  に演算  $B$  を行った後、 $Z$  基底 (computational basis) で測定すれば、上記  $\text{Tr}(M\rho)$  を測定できる。ここで、 $n$  量子ビットの状態  $\rho$  で、 $n$  個の observable 群  $\{M_k | 1 \leq k \leq n\}$  を測定することを考える。この場合、すべての  $k$  で

$$\text{Tr}(M_k\rho) = \text{Tr}[Z^{(k)}(B\rho B^{-1})] \quad \dots (2)$$

すなわち、

$$M_k = B^{-1}Z^{(k)}B \quad \dots (3)$$

が成り立つように  $B$  を決定すれば、同時測定が可能である。

( $Z^{(k)}$  は、 $k$  番目の量子ビットに対応するパウリ項が  $Z$  で、そのほかの量子ビットに対応するパウリ項がすべて  $I$  となる observable を表す。)

ここで、(3) が成立すると、任意の  $1 \leq k, l \leq n$  で

$$M_k M_l = B^{-1} Z^{(k)} Z^{(l)} B = B^{-1} Z^{(l)} Z^{(k)} B = M_l M_k \quad \dots (4)$$

であることから、 $M_k$  と  $M_l$  が可換であることが、同時測定が可能であることの必要条件になる。

また、このような演算  $B$  を施し測定する回路で、 $\{M_k\}$  内の複数の要素の積で表される observable についても期待値測定を行うことができ、最大で  $2^n - 1$  個の observable を測定することが可能になる。

### 2.2 observable 群の partitioning の方法

本研究報告の提案手法及び従来手法における partitioning は、いずれも下記のように、与えられた observable 群から clique を一つずつ生成する方法で行う。

\*\*\*\*\*

1. observable 群  $\mathcal{P}$  から、clique  $\mathcal{C}_1$  を作成。
2.  $\mathcal{P}_2 = \mathcal{P} \setminus \mathcal{C}_1$  から、clique  $\mathcal{C}_2$  を作成。
- ...
- k.  $\mathcal{P}_k = \mathcal{P}_{k-1} \setminus \mathcal{C}_{k-1}$  から、clique  $\mathcal{C}_k$  を作成。
- ...
- n.  $\mathcal{P}_n = \mathcal{P}_{n-1} \setminus \mathcal{C}_{n-1}$  がゼロ集合になる

\*\*\*\*\*

上記の流れで、observable 群  $\mathcal{P}$  を、clique  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{n-1}$  に partitioning できる。なお、 $B \setminus \mathcal{A}$  は  $B$  における  $\mathcal{A}$  の補集合を表す。

前述のように、同時測定に必要な量子回路数を小さくするためには、できるだけ clique 数を小さくする必要がある。このための一つの手段として、上記 partitioning の過程で各 clique  $\mathcal{C}_1, \mathcal{C}_2, \dots$  における observable 数を大きくする方法がある。

図2に、上記方法の例を図示している。本研究では、上記における1つの clique 生成過程で、以下の3つのアルゴリズムを適用した：(1)イジングモデルに落とし込んで DA で計算する方法 (DA アルゴリズム)、(2)Boppana-Halldórsson アルゴリズム、(3)Bron-Kerbosch アルゴリズム。これら3つのアルゴリズムの詳細を本節の以降で述べる。なお、(2),(3)については、Python モジュールである networkx [7]に格納されている機能を用いた。

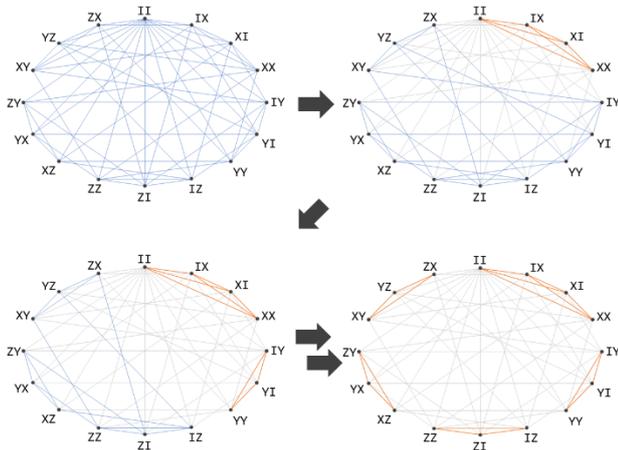


図2. 量子ビットの full tomography の場合における、対象の observable 群からの clique 形成過程。observable 間に線が結ばれることは、対象となる observable が可換であることを意味する。オレンジ色の戦で結ばれた clique が、(A) を満たす clique である。ここでは 16 個の observable を 5 つの clique に分割することができている。

### (1) DA アルゴリズム

本アルゴリズムは、observable 群  $\mathcal{X}$  から要素数が最大となる clique  $\mathcal{C}_{DA}(\mathcal{X})$  を求める問題を、イジングモデルに落としこんで DA で解くためのアルゴリズムである。以下に、その詳細を述べる：

\*\*\*\*\*

observable 群  $\mathcal{X}$  の各要素  $P_i$  に、バイナリ変数  $x_i$  を割り当てる。各変数は、 $x_i = 1$  であれば  $\mathcal{C}_{DA}(\mathcal{X})$  の要素となり、 $x_i = 0$  であれば  $\mathcal{C}_{DA}(\mathcal{X})$  の要素とならないことを意味する。

また、 $1 \leq i, j \leq |\mathcal{X}|$  を満たす各  $i, j$  で、定数  $c_{ij}$  を下記のように定義する： $P_i, P_j$  が可換であれば  $c_{ij} = 0$ 、可換でなければ  $c_{ij} = m$  とする ( $m$  は非負の定数)。これらを用いて、イジングモデル式を以下のように定義する：

$$f(\{x_i\}) = - \sum_i x_i + \sum_{i,j} c_{ij} x_i x_j \quad \dots (5)$$

(5)式の第1項は、 $x_i = 1$  となる  $x_i$  の数が大きくなるほど (つまり、 $\mathcal{C}_{DA}(\mathcal{X})$  の要素数が大きくなるほど)  $f(\{x_i\})$  が負の方向に行くことを表す。第2項は、 $x_i = x_j = 1$  かつ  $c_{ij} = m$  となるような  $\{i, j\}$  の組が多いほど (つまり、ともに  $\mathcal{C}_{DA}(\mathcal{X})$  の要素であって、かつ互いに可換でない observable の組  $(P_i, P_j)$

が多いほど)、 $f(\{x_i\})$  が正の方向に行くことを表す。ここで、 $m$  を十分大きな定数とすることで、 $f(\{x_i\})$  が最小値を取る場合の第2項の寄与をゼロにすることができる。すなわち、 $f(\{x_i\})$  が最小になる時の  $\{x_i\}$  を用いて、 $\mathcal{C}_{DA}(\mathcal{X}) = \{P_i | x_i = 1\}$  とすれば、 $\mathcal{C}_{DA}(\mathcal{X})$  を observable 群  $\mathcal{X}$  中にある要素数が最大の clique にできる。

この(5)式の最小値を、DA を用いて求める。

\*\*\*\*\*

DA は、(5)式のようなイジングモデル式の最小値を高速に計算することができる[6]。DA での計算はマルコフ連鎖モンテカルロ法に基づいており、さらに、並列試行や局所解脱出、レプリカ交換といった複数の高速化機能により、この  $2^n$  の探索空間内を効率的に探索できる ( $n$ : バイナリ変数の数)。この  $2^n$  の空間内の探索に必要なデジタルアニーラのビット数は  $n$  である。

### (2) Boppana-Halldórsson アルゴリズム[3]

本アルゴリズムは、貪欲法に準じたアルゴリズムである。まず、ある observable 群  $\mathcal{X}$  から生成する clique を  $\mathcal{C}_{BH}(\mathcal{X})$  と定義する。

$\mathcal{C}_{BH}(\mathcal{X})$  を求める方法は、以下のとおりである：

\*\*\*\*\*

まず、observable 群  $\mathcal{X}$  の要素から  $P_1$  を任意に選択する。ここで、 $P_1$  と可換な observable で構成される部分群を  $\mathcal{S}_{C(P_1)}(\mathcal{X})$  と置き、可換でない observable で構成される部分群を  $\mathcal{S}_{A(P_1)}(\mathcal{X})$  と置く。

次に、 $\mathcal{S}_{C(P_1)}(\mathcal{X})$  が零集合でない場合は、 $\mathcal{S}_{C(P_1)}(\mathcal{X})$  の要素から  $P_2$  を任意に選択する。 $\mathcal{S}_{C(P_1)}(\mathcal{X})$  の要素で  $P_2$  と可換な observable で構成される部分群を  $\mathcal{S}_{C(P_1), C(P_2)}(\mathcal{X})$ 、可換でない observable で構成される部分群を  $\mathcal{S}_{C(P_1), A(P_2)}(\mathcal{X})$  と置く。同様に、 $\mathcal{S}_{A(P_1)}(\mathcal{X})$  が零集合でない場合は、 $\mathcal{S}_{A(P_1)}(\mathcal{X})$  の要素から  $P_3$  を任意に選択する。 $\mathcal{S}_{A(P_1)}(\mathcal{X})$  の要素で  $P_3$  と可換な observable で構成される部分群を  $\mathcal{S}_{A(P_1), C(P_3)}(\mathcal{X})$ 、可換でない observable で構成される部分群を  $\mathcal{S}_{A(P_1), A(P_3)}(\mathcal{X})$  と置く。

上記の流れで、observable 群を順次分割していき、最終的にできる部分群  $\mathcal{S}_{Q_1, \dots, Q_k}(\mathcal{X})$  がすべて零集合になるようにする ( $Q_m \in \{C(P_i), A(P_i)\}, 1 \leq m \leq k$ )。

このように分解した部分群を用いて、 $\mathcal{C}_{BH}(\mathcal{X})$  を以下のようにさかのぼって定義していく。まず、 $\mathcal{S}_{Q_1, \dots, Q_m}(\mathcal{X})$  が零集合の場合は、 $\mathcal{C}_{BH}(\mathcal{S}_{Q_1, \dots, Q_m}(\mathcal{X}))$  も零集合とする。 $\mathcal{S}_{Q_1, \dots, Q_m}(\mathcal{X})$  が単一要素の場合は、 $\mathcal{C}_{BH}(\mathcal{S}_{Q_1, \dots, Q_m}(\mathcal{X})) = \mathcal{S}_{Q_1, \dots, Q_m}(\mathcal{X})$  とする。

次に、部分群  $\mathcal{S}_{Q_1, \dots, Q_{m-1}}(\mathcal{X})$  を分割するにあたって選択された要素を  $P_i$  とした場合、 $\mathcal{C}_{BH}(\mathcal{S}_{Q_1, \dots, Q_{m-1}}(\mathcal{X}))$  を  $\{P_i\} \cup \mathcal{C}_{BH}(\mathcal{S}_{Q_1, \dots, Q_{m-2}, C(P_i)}(\mathcal{X}))$  と  $\mathcal{C}_{BH}(\mathcal{S}_{Q_1, \dots, Q_{m-2}, A(P_i)}(\mathcal{X}))$  のうち、

より要素数の大きいものとする。どちらが選択されても、その要素は全て可換となる。零集合または単一要素集合からスタートして、この定義に従って順次さかのぼっていけば、すべての  $C_{BH}(\mathcal{S}_{Q_1, \dots, Q_m}(\mathcal{X}))$  を求めることができる。最終的に、 $C_{BH}(\mathcal{X})$  は  $\{P_1\} \cup C_{BH}(\mathcal{S}_{C(P_1)}(\mathcal{X}))$  と  $C_{BH}(\mathcal{S}_{A(P_1)}(\mathcal{X}))$  のうち、より要素数の大きいものとなる。

\*\*\*\*\*

### (3) Bron-Kerbosch アルゴリズム[4][5]

本アルゴリズムは、observable 群  $\mathcal{X}$  から生成する clique 群  $\{c_k(\mathcal{X})\}$  を作り、いちばん要素数の多い clique を選び、 $C_{BK}(\mathcal{X})$  とするアルゴリズムである。 $\{c_k(\mathcal{X})\}$  の作成方法を以下に示す。なお、networkx に格納されているモジュールには、clique 群  $\{c_k(\mathcal{X})\}$  を効率的に絞り込むプロトコルが組み込まれている[5]。

\*\*\*\*\*

まず、 $k = 1$  とする。

最初に  $c_k(\mathcal{X})$  を零集合とおく。 $\mathcal{X}$  から observable  $P_1$  を任意に選択し、 $c(\mathcal{X})$  に追加する。 $P_1$  と可換な observable で構成される部分群を  $N_{P_1}(\mathcal{X})$  と置く。次に、 $N_{P_1}(\mathcal{X})$  から任意の observable  $P_2$  を選択し、 $c_k(\mathcal{X})$  に追加する。 $P_2$  と可換な observable で構成される部分群を  $N_{P_2}(\mathcal{X})$  と置く。 $N_{P_1}(\mathcal{X}) \cap N_{P_2}(\mathcal{X})$  から任意の observable  $P_3$  を選択し、 $c_k(\mathcal{X})$  に追加する。この手順を、最終的に  $N_{P_1}(\mathcal{X}) \cap \dots \cap N_{P_m}(\mathcal{X})$  が零集合になるまで続け、 $c_k(\mathcal{X}) = \{P_1, \dots, P_m\}$  とする。

次に  $k \rightarrow k + 1$  とし、異なる  $P_1, P_2, \dots$  で  $c_k(\mathcal{X})$  を作成する。

上記の流れで  $c_k(\mathcal{X})$  の作成を、考えられるすべての  $P_1, P_2, \dots$  で繰り返す。

\*\*\*\*\*

## 2.3 対象となる observable 群

今回の報告では、対象となる observable 群として、full tomography や、量子化学計算 VQE 用途での partial tomography を対象とした。full tomography は高々6量子ビットのものを対象とした。VQE については、対象分子を  $H_2, LiH, H_2O, CH_4$  とし、active space 数を様々に変化させた。量子ビットへのマッピングについては、Jordan-Wigner 変換を用いた。

シミュレーション対象の observable 群を表1にまとめる。

表1. シミュレーション対象の observable 群

種類	対象分子	active spaces	量子ビット数	observable数
Full Tomography			1	4
			2	16
			3	64
			4	256
			5	1024
			6	4096
Partial tomography for VQE	H2		1	4
			2	15
	LiH		3	118
			4	193
			5	276
			6	631
	H2O		4	221
			5	312
			6	741
	CH4		7	1390
			4	241
			5	592
			6	1519
			7	3006
			8	5237

## 3. 結果と考察

### 3.1 partitioning の所要時間に関する考察

この節では、各アルゴリズムによる partitioning の所要時間を考察する。図3に、横軸を observable 数として、縦軸に partitioning の所要時間をプロットしたものを示す。observable 数  $N$  に対する本所要時間の依存性を各アルゴリズムで調べたところ、Boppana-Halldórsson アルゴリズムではおよそ  $O(N^{2.57})$ 、Bron-Kerbosch アルゴリズムではおよそ  $O(N^{5.41})$  の依存性を示した。一方 DA アルゴリズムについては、およそ  $O(N^{0.52})$  の依存性を示した。このことから、所要時間については、対象となる observable 数が大きいほど、DA アルゴリズムにアドバンテージがあると期待される。

今回のシミュレーション結果では、Bron-Kerbosch アルゴリズムに対しては 300 以上の observable 数で、Boppana-Halldórsson アルゴリズムに対しては 2000 以上の observable 数で、それぞれ DA アルゴリズムの方が高速に partitioning できている (図 3b)。特に observable 数が 5237 の場合、Boppana-Halldórsson アルゴリズムでは 1 時間程度かかるところを、DA アルゴリズムでは 7 分程度で partitioning ができている。

ここで、DA アルゴリズムにおける所要時間について詳しく調査するため、生成した clique 数に対する partitioning の所要時間の依存性を調べた (図 4)。その結果、それらは比例関係にあることが分かった。これは、1 つの clique 生

成にかかる時間が一定（約 4 秒）であることによると考えられる。このことから、DA アルゴリズムにおける所要時間の依存性は、observable 数に対する、最終的に生成する clique 数の依存性を反映したものと考えられる。

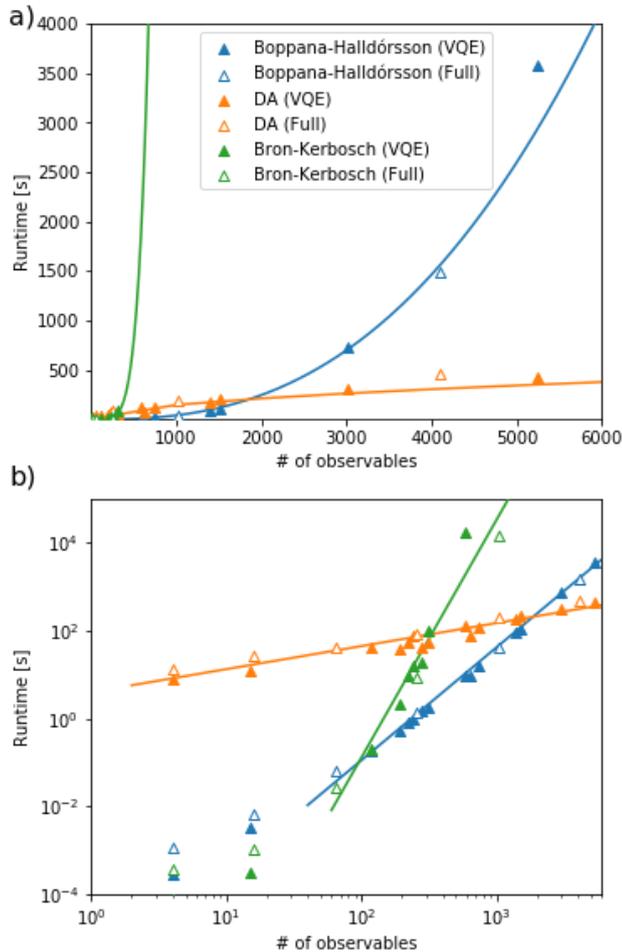


図 3 a) observable 数に対する、各アルゴリズムによる partitioning の所要時間のプロット。b) プロット a の両軸を対数軸としたもの

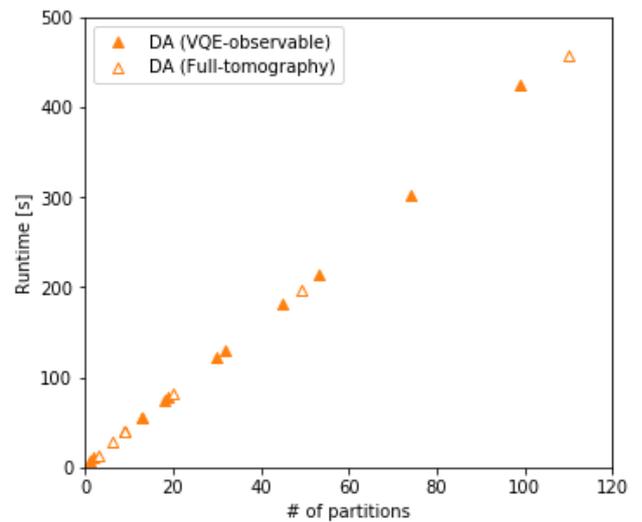


図 4 DA アルゴリズムにおける partitioning の所要時間の、生成 clique 数に対するプロット

### 3.2 clique 数に関する考察

本節では、2.2 節で示した各アルゴリズムにより生成した clique 数について考察を行う。図 5、図 6 に、各オブザーバブル群に対して生成した clique 数を示す。オブザーバブル数が 20 以下の時は、どのアルゴリズムでもほぼ同じ clique 数になった。一方 20 を超えると、Boppana-Halldórsson アルゴリズムより、DA アルゴリズムや Bron-Kerbosch アルゴリズムの方がより少ない clique 数の結果となった。これは、DA アルゴリズムや Bron-Kerbosch アルゴリズムが、1 回の clique の作成過程でその要素数が最大になるようなアルゴリズムになっているのに対し、Boppana-Halldórsson アルゴリズムは、近似が組み込まれている点から必ずしも要素数が最大になっていないことによると考えられる。この点は、図 7 から確認ができる。DA-アルゴリズムや Bron-Kerbosch アルゴリズムでは、要素数が多いものから clique が順に生成しているのに対し、Boppana-Halldórsson アルゴリズムでは、後に生成した clique の方が要素数の大きいことがある。

なお、DA アルゴリズムと Bron-Kerbosch アルゴリズムで clique 数が必ずしも同じになっていない。これは、1 回の clique 生成過程で（要素数が最大の）clique が複数考えられる場合があり、その中でどの clique を選択したか、に起因すると推察される。

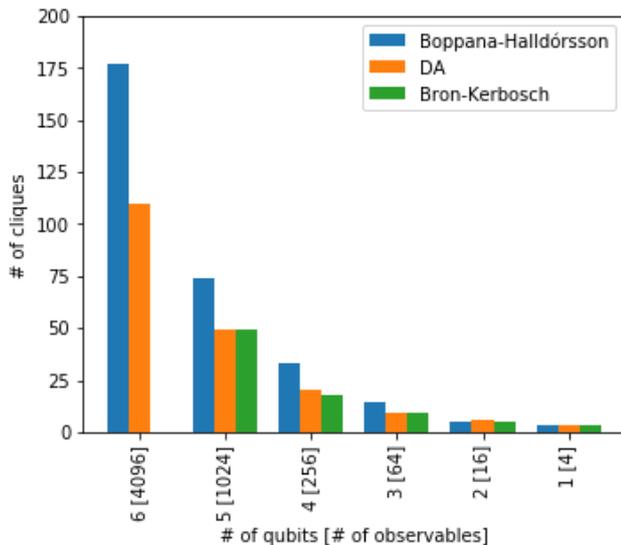


図 5 full tomography を想定して求めた clique 数の、各手法間での比較。Bron-Kerbosch アルゴリズムによる 6 量子ビットのシミュレーションは、6 時間以上を要するため、シミュレーション対象としなかった。

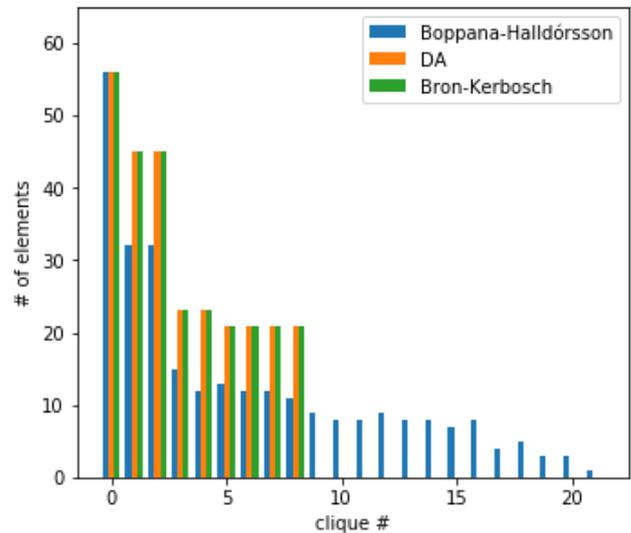


図 7 276 個の observable (VQE による partial tomography で、LiH 分子を active space=5 でシミュレーションする場合) を partitioning した時の各 clique 内の observable 数

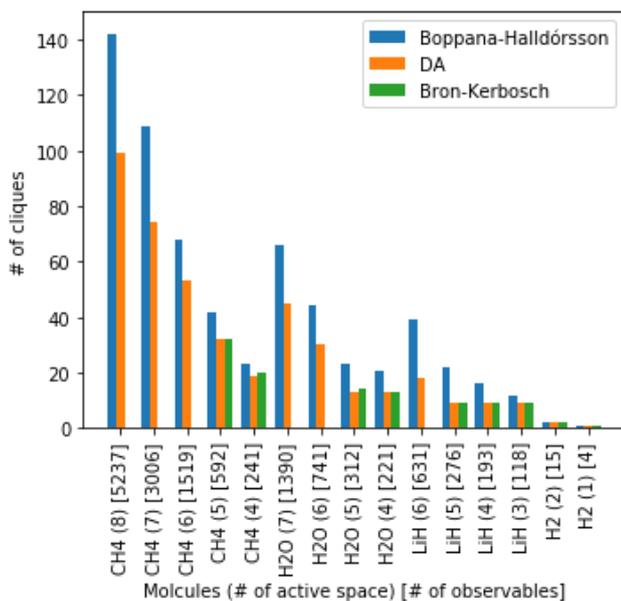


図 6 VQE の各設定条件で求めた clique 数の、各手法間での比較。Bron-Kerbosch アルゴリズムについては、シミュレーションに 6 時間以上を要するものは、シミュレーション対象としなかった。

#### 4. まとめ

同時測定において重要なステップとなる partitioning を、イジングモデルに落とし込んで DA で解く方法を提案した。従来手法である Boppana-Halldórsson アルゴリズムや Bron-Kerbosch アルゴリズムと比較した結果、observable 数が多くなると、clique 数と所要時間の両方で提案手法にアドバンテージがあることを見出した。このことから、本方法は、今後量子コンピュータが大規模化した場合に利用できる partitioning 手法の一つになると期待される。また、今回提案した partitioning 手法はアルゴリズムの種類を選ばないため、汎用性が高い方法となることも強調しておきたい。

一方、同時測定を効率的に行うためには、本 partitioning に加え、図 1 のステップ④で示すような基底変換ゲート  $B$  を量子回路に実装する必要がある。この実装方法は簡単ではなく、今後の研究対象になると考えられる。現時点では、例えば文献[2]に記載されている方法がある。この方法を組み合わせることで、量子計算の速度が向上することが期待される。

なお、今回は 8,192 ビットを有する第二世代の DA を用いて計算を行った[6]。本報告の partitioning の例は全て第二世代 DA の規模の範囲内で扱えるものである。一方、さらなる大規模化のニーズを見据えて、メガビット (1Mbit) 規模の求解を可能とする DA の大規模求解システムの開発も進められている[8]。測定対象成分の数が更に多い場合の計算時間のスケールリングや clique 数の比較については、今後の検討課題としたい。

## 参考文献

- [1] Verteletskyi, V. Yen, T.-C. Izmaylov, A. F. “Measurement optimization in the variational quantum eigensolver using a minimum clique cover” *J. Chem. Phys.* **152**, 2020, 124114
- [2] Gokhale, P. et al. “ $O(N^3)$  Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians” *IEEE Transactions on Quantum Engineering*, **1**, 2020, 1-24, Article no: 3102324
- [3] Boppana, R. and Halldórsson, M. “Approximating Maximum Independent Sets by Excluding Subgraphs” *BIT Numerical Mathematics*, **32**, 1992, 180–196
- [4] Bron, C. and Kerbosch, J. “Algorithm 457: finding all cliques of an undirected graph”. *Communications of the ACM* **16**, **9**, 1973, 575–577
- [5] Tomita, E. Tanaka, A. Takahashi, H. “The worst-case time complexity for generating all maximal cliques and computational experiments”. *Theoretical Computer Science* **363**, 2006, 28-42
- [6] 竹本一矢、松原聡、渡部康弘、島田大地、栗田知周、田村泰孝：“組合せ最適化問題を高速に解く「デジタルアニーラ」技術と製造・材料科学への応用”、電子情報通信学会論文誌C, **J104-C(4)**, 2021, 101-109
- [7] <https://networkx.org/>
- [8] <https://pr.fujitsu.com/jp/news/2020/11/9.html>