量子コンピューティングへの自動チューニングの適用と評価

森下誠[1]　片桐孝洋[2]　大島聡史[2,3]　星野哲也[2]　永井亨[2]

概要：本報告は，量子コンピュータおよび関連技術における自動チューニング(AT)技術の適用の有用性を調査する．また，解の高精度化および実行時間の高速化を目的とする．本報告では量子関連技術に焦点を当て，量子インスパイア型のアニーリング方式で用いられる QUBO 式中の制約項の係数や，ゲート方式の GPU シミュレータにおけるスレッド数などをチューニング対象のパラメータとする．AT 適用の前段階として，本報告ではチューニング対象のパラメータが解の精度や実行時間にどの程度影響するのかを調査した．実験結果から，チューニング対象のパラメータが解の精度や実行時間に明らかに影響を及ぼすことを確認し，その影響が問題サイズによって異なることが分かった．

キーワード：CMOS アニーリングマシン，cuQuantum，自動チューニング

1. はじめに

次世代の計算機として注目されている量子コンピュータは，冷却装置のコストが大きく，技術的に量子ビット数が限られるなど，実用化の観点からは課題が残る．そのため，現状では量子コンピュータそのものではなく，量子関連技術としての量子インスパイア型イジングマシンや，量子回路シミュレータの実用化を目指す動きがある．

量子インスパイア型イジングマシンとは，量子コンピュータの動作原理である量子効果をデジタル回路で模倣することで，常温で動作可能であり，低コストかつ高スケーラビリティを実現した計算機である．その 1 つとして，日立製作所の山岡らにより, CMOS アニーリングマシン[1][2]が提案されている．

日本では，日立，富士通，東芝の 3 社によって量子インスパイア型アニーリングマシンの研究・開発が行われている．アニーリングマシンはいくつかの組合せ最適化問題において，従来の計算機での実行に対する優位性を示すことが期待されている．例えば，組合せ最適化問題の中でもクラスタリングは多岐に渡る応用があるため，量子コンピュータや関連技術に適用が期待されている．

著者らは数値計算プログラムを中心に，プログラム上の性能パラメータを自動チューニングする技術である，ソフトウェア自動チューニング（AT）の研究を行ってきた[3]. AT が取り扱う問題では，教師あり学習，かつ，分類問題となることが多いため，クラスタリングが AT に利用できると予想される．また，AT で必要とされるデータは一般に大規模となることが多く，性能向上の観点からクラスタリングの高速化，および高精度化が求められる．そのため，例えば CMOS アニーリングマシンによりクラスタリングが高速化されれば，AT の適用が期待できる．我々は，前回の報告[4]において，最小頂点被覆問題での CMOS アニーリングマシンの性能評価を行ってきた．

本研究は，量子コンピュータおよび関連技術における自動チューニング技術の適用の有用性を調査し，解の高精度化および実行時間の高速化を目的とする．本報告では量子関連技術に焦点を当て，量子インスパイア型のアニーリング方式で用いられる QUBO 式中の制約項の係数や，ゲート方式の GPU シミュレータにおけるスレッド数などをチューニング対象のパラメータとする．

本報告の構成は以下のとおりである．2 章でアニーリング方式における自動チューニング対象，3 章でゲート方式における自動チューニング対象について解説する．4 章は，2 章および 3 章で登場した性能パラメータを変化させた場合にどの程度の性能変動があるかを示す．最後に 5 章で本報告のまとめを行う．

2. アニーリング方式での自動チューニング

アニーリングマシンでは，頂点と辺で構成される無向グラフであるイジングモデルによって問題を表現する．マシンによってそのグラフの構造が異なるため，実装の際にはグラフの特徴を理解することが必要である．表 1 に代表的なアニーリングマシンの一覧を示す．

表 1　代表的なアニーリングマシン

名称	開発会社	実装方式	搭載ビット数	計算グラフ
D-Wave Advantage [5]	D-Wave Systems	QPU	5,000	ペガサスグラフ
CMOS アニーリングマシン	日立製作所	専用回路 GPU	147,456 262,144	キンググラフ
FujitsuDA3Solver（Digital Annealer）[6]	富士通	専用回路	100,000	完全グラフ
SBM PoC 版（シミュレーテッド分岐マシン）[7]	東芝	FPGA	10,000	完全グラフ

1 名古屋大学 大学院情報学研究科
2 名古屋大学 情報基盤センター
3 九州大学 情報基盤研究開発センター

ここで，実装方式はアニーリングを実現するために用いるハードウェア，搭載ビット数は計算に利用可能な量子ビットに相当する数，計算グラフはアニーリングマシンでイジングモデルを表現するために用いるグラフを表している．

表 1 からわかるように，CMOS アニーリングマシンなどの量子インスパイア型イジングマシンは，量子アニーリングマシンである D-Wave Advantage よりも計算ビットを多く利用できるという利点がある．また，CMOS アニーリングマシンの計算グラフとして採用されているキンググラフは，完全グラフと比較すると最適化問題の埋め込みに工夫が必要となるものの，疎結合グラフであるためスケールが可能である．一方，完全グラフは最適化問題の埋め込みが容易であるという特徴がある．

図 1 にアニーリングマシンで最適化問題を解く場合の処理手順を示す．

1.最適化問題

最初に解きたい最適化問題を選定する．アニーリングマシンは組み合せ最適化問題や二次計画問題などのイジングモデルで表現可能な最適化問題を解くことができる．

2.イジングモデル

アニーリングマシンは，入力としてイジングモデルと呼ばれる計算モデルのパラメータを受け取る必要があるため，最適化問題をイジングモデルで表現する．イジングモデルは（1）式のエネルギー関数で表される．

$$H = \sum_{i \neq j} J_{ij}\sigma_i\sigma_j + \sum_i h_i\sigma_i \qquad (1)$$

ここで，σ_iはスピン状態，J_{ij}は相互作用（二体のパラメータ），h_iは磁場（一体のパラメータ）を表し，$\sigma_i \in \{-1, +1\}$である．アニーリングマシンを最適化問題に適用する際は，マシンへの入力として相互作用と磁場を与え，最適化問題の解としてスピンの状態を出力として受け取る．イジングモデルの表現に用いる計算グラフでは，相互作用が計算グラフの辺に付与されるパラメータ，磁場が計算グラフの頂点に付与されるパラメータとなる．

3.マシン実行

アニーリングマシンの実行は，実機を直接実行する場合と API (Application Programming Interface) を介して実行する場合の 2 通りがある．いずれもマシンへの入力として磁場と相互作用のパラメータを与える．

4.スピン取得

アニーリングマシンの実行後に座標毎のスピン状態が得られる．このスピン状態の集合が最適化問題の解に対応する．

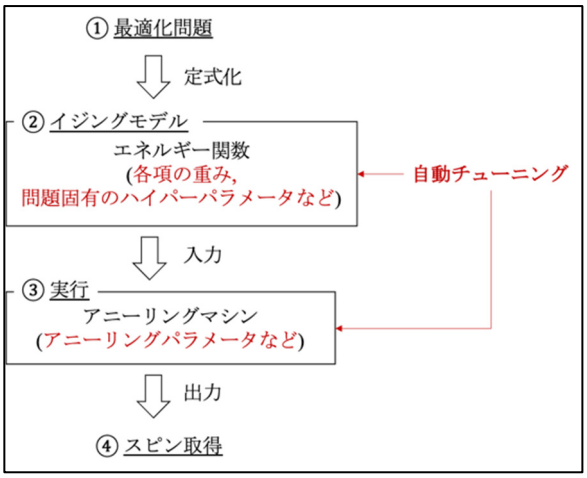

図 1　アニーリング方式の処理手順

図 1 の赤字で示すように，アニーリング方式の処理中に出てくるチューニング対象のパラメータとしては，イジングモデルのエネルギー関数における各項の重みや，アニーリング実行時の初期温度などのアニーリングパラメータなどが挙げられる．

具体例として，（2）式で表される最小頂点被覆問題を CMOS アニーリングマシンで解く際のチューニング対象となるパラメータを表 2 に示す．

$$H = w_a \sum_{(u,v)\in E} (1 - x_u)(1 - x_v) + w_b \sum_{v \in V'} x_v \qquad (2)$$

表 2　最小頂点被覆問題を CMOS アニーリングマシンで解く際のチューニング対象となるパラメータ

パラメータ名	説明
Wa	制約項の重み
Wb	最適化項の重み
chain_strength	チェインの強さ
temperature_num_steps	アニーリングのステップ数
temperature_step_length	アニーリングのステップ長
temperature_initial	アニーリングの初期温度
temperature_target	アニーリングの最終温度

ここで，w_a, w_bは 0 より大きい定数であり，第一項が制約項，第二項が最適化項である．また，$x \in \{0,1\}$はバイナリ変数，Eは問題グラフの辺の集合，V'は問題グラフの頂点被覆集合を表す．なお，無向グラフにおいて，全ての枝$e \in E$の少なくとも一方が$V' \in V$に含まれているとき，V'を頂点被覆と呼ぶ．最小頂点被覆問題とは，頂点被覆集合の要素数$|V'|$が最小となるV'を求める問題である．

図 2 に示す 5 頂点のグラフでは，赤く塗られた 2 頂点が最小頂点被覆となる例である．

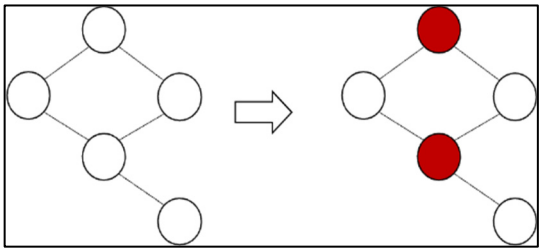

図 2　最小頂点被覆問題の例

3. ゲート方式での自動チューニング

ゲート方式のマシンでは，解く問題を量子回路モデルに落とし込んで問題を解く．図 3 にゲート方式で問題を解く場合の処理手順を示す．

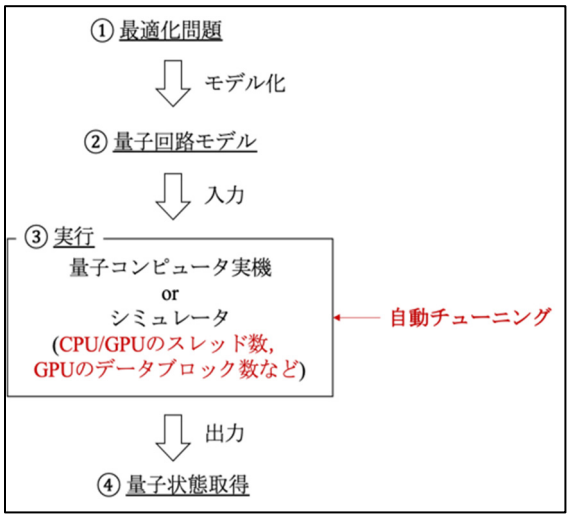

図 3　ゲート方式の処理手順

1.最適化問題

アニーリング方式と同様に最初に解く問題を選定する．こちらの方式も解の状態はスピン状態として得られるため，解が 0 と 1 の組み合わせで表現できる 3-SAT などの組み合せ最適化問題などを解くことができる．

2.量子回路モデル

ゲート方式では，解く問題毎に量子回路モデルを考える必要がある．これはアニーリング方式ではイジングモデルに相当する．多くの量子回路モデルは，量子状態の重ね合わせで全て解の状態を網羅し，ゲート操作を加えることによって，目的の解の状態のみの観測確率を上げるようなモデルになっている．

3.実行

ゲート方式の実行は実機を実行する場合とシミュレータを実行する場合の 2 通りがある．シミュレータ実行の場合は，ライブラリによっては GPU (Graphics Processing Unit)を用いてより高速な実行が可能である．

4.量子状態取得

実行後には各量子状態の観測確率が得られる．この観測確率の高い量子状態に対応する解が，問題の解となる．

図 3 の赤字で示すように，ゲート方式の処理中に出てくるチューニング対象のパラメータとしては，量子回路シミュレータに限定すると，実行時の CPU/GPU スレッド数やデータブロック数などが挙げられる．

具体例として，cuQuantum [8]で量子シミュレーションを解く際のチューニング対象となるパラメータを表 3 に示す．

表 3　cuQuantum で量子シミュレーションを解く際のチューニング対象となるパラメータ

パラメータ名	説明
max_fused_gate_size	fused_gate 毎の最大量子ビット数
cpu_threads	CPU 実行時に使用するスレッド数
use_gpu	GPU を使うかどうか
gpu_mode	値が 0 の場合に CUDA，それ以外の値の場合に cuStateVec ライブラリを使用
gpu_state_threads	CUDA ブロック毎のスレッド数
gpu_data_blocks	GPU で使用するデータブロック数

4. 性能評価

4.1 実験環境

本報告での，アニーリング方式，ゲート方式の実験環境をそれぞれ表 4，表 5 に示す．

表 4　アニーリング方式の実験環境

利用マシン及びライブラリ	説明
CMOS アニーリングマシン [9]	• Annealing Cloud Web API v2 を使用 • マシンタイプは GPU 版（32bit / float）
MacBookAir（macOS Big Sur）	• Python 実行用のマシン • CPU : 1.6 GHz，2 Core，Intel Core i5 • メモリ : 8 GB
Amplify [10]	• マシン利用のためのライブラリ • Version 0.5.13

表 5　ゲート方式の実験環境

利用マシン及びライブラリ	説明
スーパーコンピュータ「不老」Type II サブシステム [11]（1 ノード）	• FUJITSU Server PRIMERGY CX2570 M5 • CPU：2.10-3.90 GHz，20 core，Intel Xeon Gold 6230 • GPU：NVIDIA Tesla V100 SXM2 • メモリ：384 GiB
cuQuantum	• 量子回路シミュレータ実行用のマシン • NVIDIA 製の SDK

4.2 実験結果

4.2.1 アニーリング方式

一辺の長さ$N = 7,8$の正方格子グラフの最小頂点被覆問題において，チューニングパラメータw_a, w_b, chain_strength を，それぞれ0.0~2.0の範囲で変化させた時の精度変化を図 4～9 に示す．

なお，解の精度の評価指標として最適解解答率を用いている．本実験ではアニーリングマシンの実行回数を 10 回とした．

$$最適解解答率 = \frac{最適解が得られた回数}{アニーリングマシンの実行回数} \quad (3)$$

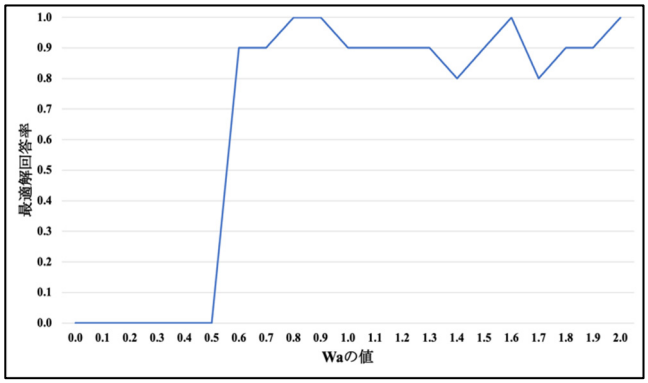

図 4　一辺の長さ N=8，$w_b = 1.0$，chain_strength=1.0 の最小頂点被覆問題において，$w_a = 0.0$~2.0と変化させた時の最適解解答率の推移

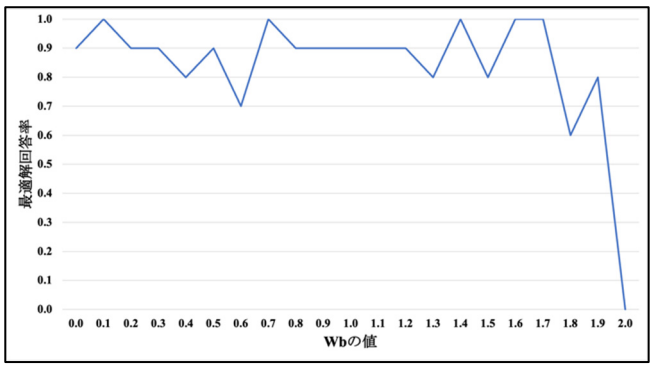

図 5　一辺の長さ N=8，$w_a = 1.0$，chain_strength=1.0 の最小頂点被覆問題において，$w_b = 0.0$~2.0と変化させた時の最適解解答率の推移

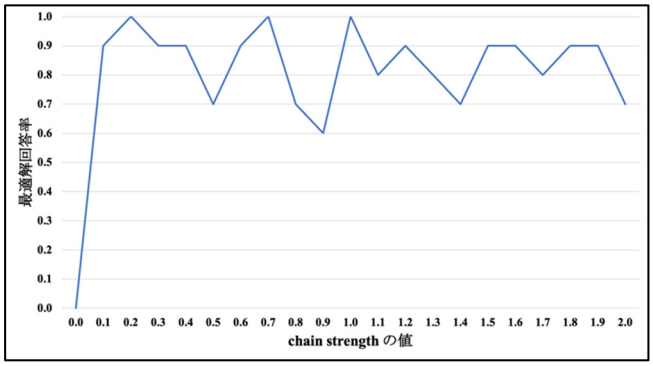

図 6　一辺の長さ N=8，$w_a = 1.0$，w_b=1.0 の最小頂点被覆問題において，chain_strength ＝0.0~2.0と変化させた時の最適解解答率の推移

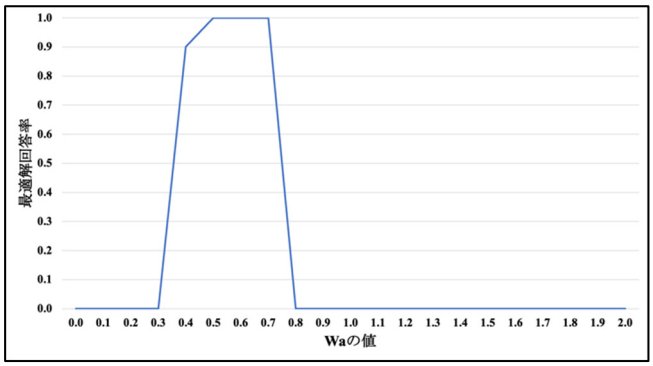

図 7　一辺の長さ N=7，$w_b = 1.0$，chain_strength=1.0 の最小頂点被覆問題において，$w_a = 0.0$~2.0と変化させた時の最適解解答率の推移

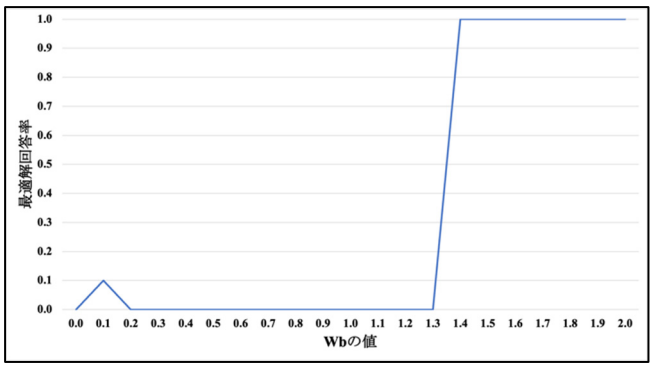

図 8　一辺の長さ N=7，$w_a = 1.0$，chain_strength=1.0 の最小頂点被覆問題において，$w_b = 0.0$~2.0と変化させた時の最適解解答率の推移

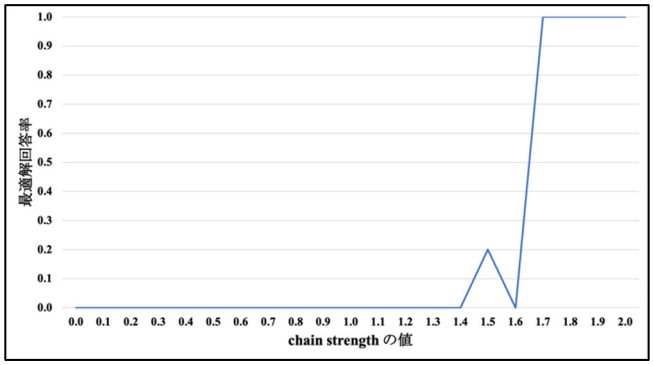

図 9　一辺の長さ N=7，$w_a = 1.0$，w_b=1.0 の最小頂点被覆問題において，chain_strength ＝0.0~2.0と変化させた時の最適解解答率の推移

以上の結果より，パラメータ 1 つを取っても，アニーリングマシンの解の精度に影響があることは明らかである．

図 4，図 7 を比較すると，N=8 では$0.5 < w_a$で 80[%]以上の最適解解答率を得られているが，N=7 では$0.3 < w_a < 0.8$で最適解回答率 90[%]以上となっている．w_bや chain_strength の値を変化させたときも同様のことが起きていることから，同じ問題においても問題サイズが変わるとパラメータの最適値が変化するということがわかる．

また N=8 と比べて全体的に N=7 の最適解回答率が低いのは，最小頂点被覆問題の最適解をアニーリングマシンで導くに当たって，一辺の長さが奇数の正方格子グラフの問題が偶数のものよりも難しいことに起因する．これは，奇数の問題が最適解と非常に近しい準最適解を持つため，アニーリングマシンでは後者の局所解で安定してしまうことがあるからである．

4.2.2 ゲート方式

ここでは，杉崎らにより提供された，波動関数の時間発展量子シミュレーションのベンチマーク（以降，量子シミュレーションベンチマーク）を用いて，性能評価を行った．ここでは，量子ビット数=8, 10, 16, 18 の波動関数の時間発展量子シミュレーションにおいて，チューニングパラメータ max_fused_gate_size を 2, 3, 4，gpu_mode を 0, 1 の範囲で変化させた時の実行時間推移を図 10〜13 に示す．

本実験の評価指標としては，シミュレーションの実行時間を用いた．

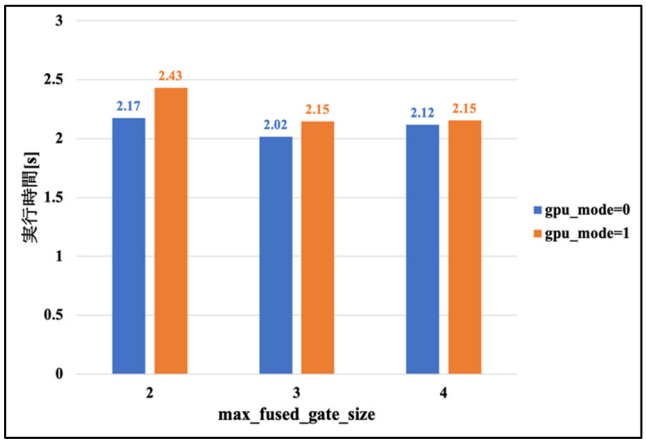

図 10　量子ビット数=8[bit]の量子シミュレーションベンチマークで max_fused_gate のサイズと gpu_mode を変更した時の実行時間の推移

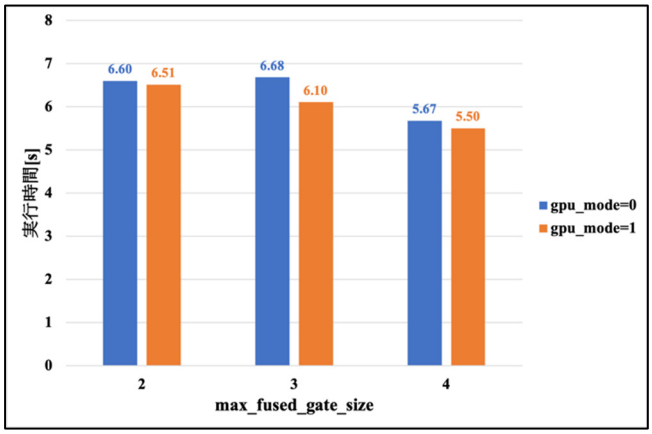

図 11　量子ビット数=10[bit]の量子シミュレーションベンチマークで max_fused_gate のサイズと gpu_mode を変更した時の実行時間の推移

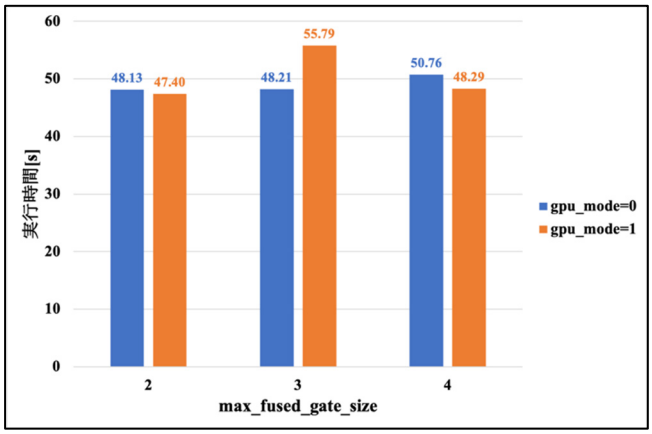

図 12　量子ビット数=16[bit]の量子シミュレーションベンチマークで max_fused_gate のサイズと gpu_mode を変更した時の実行時間の推移

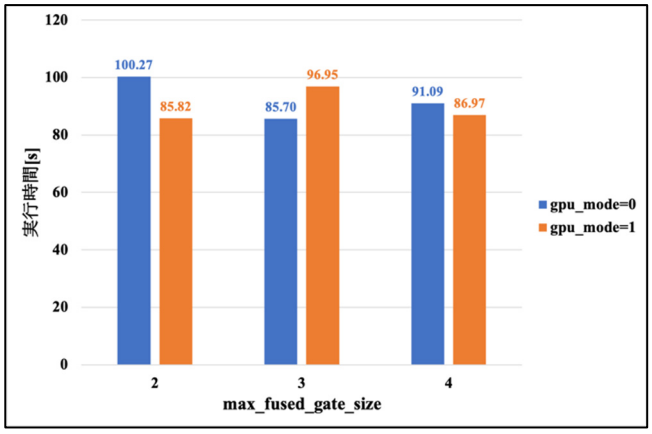

図 13　量子ビット数=18[bit]の量子シミュレーションベンチマークで max_fused_gate のサイズと gpu_mode を変更した時の実行時間の推移

それぞれの結果で，最も実行時間が短いものとデフォルト実行（gpu_mode=0，max_fused_gate_size=2） の実行時間を比べると，表 6 のようになる．

表 6　デフォルト実行時間と最短実行時間の比較

$$\left(高速化率 = \frac{デフォルト実行時間}{最短実行時間}\right)$$

量子ビット数[bit]	デフォルト実行時間[s]	最短実行時間[s]	高速化率
8	2.17	2.02	1.07
10	6.60	5.50	1.20
16	48.13	47.40	1.01
18	100.27	85.70	1.17

表 6 より，どの量子ビット数においても，チューニングパラメータを変動させたときに実行時間が変化している．また，10[bit]のシミュレーションにおいて，デフォルト実行時間と最短実行時間の比率（高速化率）が最も大きい 1.20 倍となった．

一方，パラメータを変動させると性能が悪くなる場合もある．例えば，16[bit]のシミュレーションにおいて，max_fused_gate_size=3, gpu_mode=1 の場合は高速化率=0.86 となった．

5. おわりに

本報告では，CMOS アニーリングマシンによって正方格子グラフ上の最小頂点被覆問題，cuQuantum によって量子シミュレーションを実行し，それぞれ解の精度と実行時間を評価した．

実験の結果，チューニング対象のパラメータが，解の精度や実行時間に明らかに影響を及ぼすことを確認し，その影響は問題サイズによって異なることが分かった．このことから，この性能パラメータに対する AT の必要性があるといえる．

本実験ではベンチマーク問題として最小頂点被覆問題と波動関数の時間発展シミュレーションの 2 つを用いた．これらは，どちらもイジングモデルや量子回路モデルといった一般的な計算モデルを使用しているため，本実験のような結果が生じることは特殊ではなく，別の問題を扱う際にも生じると予測される．そのため，この両モデルを取り扱う際に現れる性能パラメータのチューニングは重要な課題であると考える．

今後はアニーリング方式，および，ゲート方式の両方式の計算機へ AT 技術を適用する方式を検討する必要がある．例えば著者らは，反復解法の前処理選択の問題において，疎行列の形状，非零要素数，および，行列サイズの情報を画像化して機械学習をさせることで，AI 技術を AT 技術に適用させる手法[12]の提案を行っている．同様に，量子回路シミュレーションの特徴を画像化し，性能に関するパラメータの最適化を AI にさせることで，新しい AT 機構の開発が可能かもしれない．これらの性能パラメータのチューニングを自動化する AT 機構の開発と，更なる性能評価を行っていく予定である．

謝辞　本研究は JSPS 科研費　JP19H05662 の助成を受けたものです．

CMOS アニーラの利用に関して助言を頂いた，日立製作所の小埜和夫氏と山岡雅直氏に感謝します．

Amplify の利用に関して助言を頂いた，株式会社フィックスターズの松田佳希氏に感謝します．

cuQuantum による量子回路シミュレーションの評価に関して，助言およびベンチマーク提供を頂いた，大阪公立大学の杉崎研司 特任講師, および 立教大学の望月祐志 教授に感謝いたします．

参考文献

[1] M. Yamaoka, “A 20k-spin Ising chip for combinatorial optimization problems with CMOS annealing, “IEEE International Solid-State Circuits Conference, 2015.
[2] 山岡雅直，“組合せ最適化問題に向けた CMOS アニーリングマシン”，電子通信学会 基礎・境界ソサイエティ，Fundamentals Review, Vol.11, No.3, pp.164-171, 2018.
[3] T. Katagiri and D. Takahashi, “Japanese Autotuning Research: Autotuning Languages and FFT”, Proc. of the IEEE, Vol. 106, Issue 1, pp. 2056-2067, 2018.
[4] 森下誠，片桐孝洋，大島聡史，永井亨, Amplify を用いた CMOS アニーリングマシンの特性の分析，研究報告ハイパフォーマンスコンピューティング（HPC）, Vol. 2021-HPC-181, No. 3, pp. 1-6, 2021.
[5] Bhatia, Harshil Singh, and Frank Phillipson. ”Performance Analysis of Support Vector Machine Implementations on the D-Wave Quantum Annealer.” International Conference on Computational Science. Springer, Cham, 2021.
[6] Sao, Masataka, et al. ”Application of digital annealer for faster combinatorial optimization.” Fujitsu Scientific and Technical Journal, Vol. 55, No.2, pp.45-51, 2019.

[7] Goto, Hayato, Kosuke Tatsumura, and Alexander R. Dixon. ”Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems.” Science advances 5.4, 2019.
[8] cuQuantum https://developer.nvidia.com/cuquantum-sdk (閲覧日:2023 年 2 月 13 日).
[9] CMOS アニーリングマシン – Annealing Cloud Web https://annealing-cloud.com (閲覧日:2023 年 2 月 13 日).
[10] Amplify – 量子アニーリングと共に進化するクラウド https://amplify.fixstars.com/ja/ (閲覧日:2023 年 2 月 13 日).
[11]名古屋大学情報基盤センター，スーパーコンピュータ「不老」TypeⅡサブシステム https://icts.nagoya-u.ac.jp/ja/sc/overview.html#type2 (閲覧日:2023 年 2 月 13 日).
[12] Kenya Yamada, Takahiro Katagiri, Hiroyuki Takizawa, Kazuo Minami, Mitsuo Yokokawa, Toru Nagai, Masao Ogino, Preconditioner Auto-tuning with Deep Learning for Sparse Iterative Algorithms, Proceedings of CANDAR18, pp. 1-8, 2018.