

Amplify を用いた CMOS アニーリングマシンの特性の分析

森下誠¹ 片桐孝洋² 大島聡史² 永井亨²

概要: 本研究は、量子コンピュータを筆頭とする次世代計算機の活用のため、量子インスパイア型イジングマシンの1つである CMOS アニーリングマシンを典型的な問題に適用する際の課題を明らかにすることで、より実用的な問題への応用を目指す。具体的には、アニーリングマシンを取り扱うための python ライブラリである Amplify を利用し、エネルギー関数の重みの変更による出力の変化を調べた。実験結果から、エネルギー関数中の最適化項に対する制約項の係数値が 0.375~1.5 倍辺りのとき、CMOS アニーリングマシンが最適解を導く確率が高いことが分かった。

キーワード: CMOS アニーリングマシン, Amplify

1. はじめに

1.1 CMOS アニーリングマシンと自動チューニング

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、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、量子インスパイア型イジングマシンが注目されている。量子インスパイア型イジングマシンの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]が提案されている。

日本では、日立、富士通、東芝の3社によってアニーリングマシンの研究・開発が盛んに行なわれている。アニーリングマシンはいくつかの組み合わせ最適化問題において、従来の計算機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。

著者らは、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)[2]の研究を行っている。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の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及び高精度化が求められる。そのため、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への適用が期待できる。

1.2 CMOS アニーリングマシンの適用

CMOS アニーリングマシンは、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(図1)。

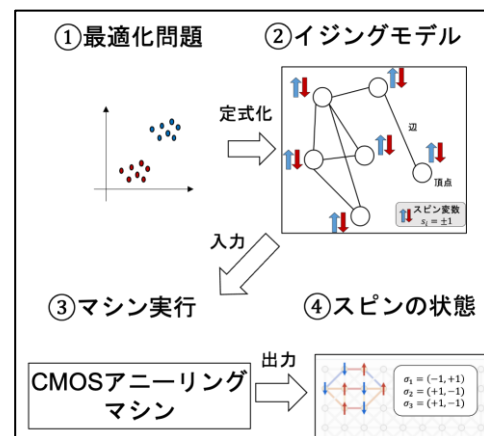


図1 CMOS アニーリングマシンによる求解の手順

そのため、最適化問題のグラフ帰着に伴う問題が生じる。

1 名古屋大学大学院情報学研究科
Graduate School of Informatics, Nagoya University
2 名古屋大学 情報基盤センター
Information Technology Center, Nagoya University

特に、本研究で対象とする CMOS アニーリングマシンにおける問題点を以下に示す。

- ① クラスタリング問題や巡回セールスマン問題などの組合せ最適化問題は、内部的にイジングモデルで表現する際に全結合マッピングが必要となる。疎結合なグラフ構造を持つ CMOS アニーリングマシンでは、グラフのノードを複製して全結合を行う「ノードチェーン」が用いられる。しかしながら、このノードチェーンは、単純な方法ではマッピングの効率が悪くなるため、効率化する手法の開発[3]がされている。
- ② ノード間の相互作用の値の適切な大きさは、解く問題のサイズやマシン固有の構造によって決定される。相互作用の値が小さすぎたり大きすぎたりすると、解が破綻する場合がある。

本研究では 2 つ目の課題に着目する。CMOS アニーリングマシンの特性を分析することで、一般的なアニーリングマシンに関連するパラメータ設定の自動化を行い、適用局面の拡大を図る。

本報告の構成は以下のとおりである。2 章で最小頂点被覆問題を解説する。3 章は、実験環境の Amplify について説明する。4 章で、Amplify を利用し、マシンのパラメータやエネルギー関数の重みの変更によるマシンの出力の変化を調べる予備評価を行う。最後に 5 章で本報告のまとめを行う。

2. 最小頂点被覆問題

2.1 概要

無向グラフ $G = (V, E)$ 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。頂点被覆となる頂点集合の要素数 $|V'|$ が最小となる V' を求める問題が最小頂点被覆問題である。

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。

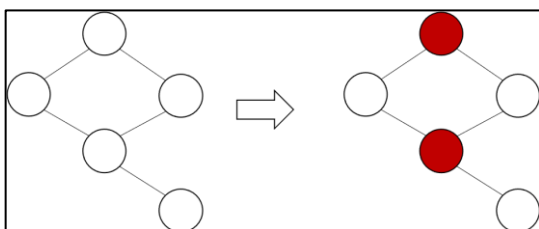


図 2 最小頂点被覆問題の例

本研究では、図 3 に示す正方格子グラフにおける最小頂点被覆問題を解く。図 3 の問題では、1 辺の長さが偶数の正方格子グラフとなっているため、理論的な最適解がわかる。この最適解では、最小頂点被覆集合の要素数が総頂点数の半分となる。理論的な最適解がわかるため、本研究の予備評

価の解の精度の検証に用いる。

2.2 評価指標

ここで、解に対する評価指標である最適解回答率を式 (1) で定義する。

$$\text{最適解回答率} = \frac{\text{最適解回答数}}{\text{計算回数}} \times 100[\%] \quad \dots (1)$$

式 (1) の計算回数とは、評価対象の CMOS アニーリングマシンを実行する回数とする。これは図 1 中の③を繰り返すことに相当し、イジングモデルは Amplify によってエネルギー関数としてあらかじめ定義しておく。また最適解回答数は、最小頂点被覆に対応する最適解が得られた回数である。ここでは、準最適解と、解が返されない場合とは区別せず、どちらも最適解回答には含めないものとする。

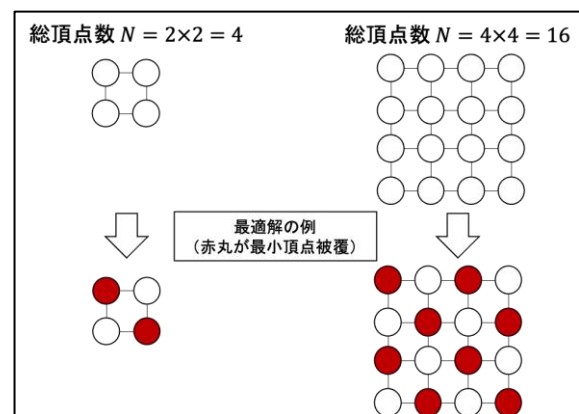


図 3 正方格子グラフにおける最小頂点被覆問題

3. Amplify による実装

3.1 概要

本研究では、CMOS アニーリングマシンを用いた実験で最小頂点被覆問題を解くにあたり、Amplify [4] を利用した。Amplify は、株式会社フィクスターズが開発しているイジングマシン向けミドルウェアライブラリである。

イジングマシンとは、二値変数の二次多項式で記述される最適化問題の求解に特化したマシンで、CMOS アニーリングマシンもこれに含まれる。Amplify では、イジングマシンを使用する際のパラメータ取得と設定、マシンの実行を行うインターフェースを提供している。これにより、ユーザーの目的に応じた柔軟なマシン操作が可能となる。

3.2 エネルギー関数の実装

アニーリングマシンで最適化問題を解く際には、最適化問題をバイナリ変数もしくはイジング変数を用いてエネルギー関数として定式化する必要がある。

バイナリ変数 $x \in \{0,1\}$ を使った場合、最小頂点被覆問題のエネルギー関数は式(2)で表される。

$$H = A \sum_{(u,v) \in E} (1 - x_u)(1 - x_v) + B \sum_{v \in V'} x_v \quad \dots (2)$$

ここで、 A, B は 0 より大きい定数であり、第 1 項が制約項、第 2 項が最適化項である。このエネルギー関数である式(2)の Amplify による実装例を図 4 に示す。

```
# QUBO 変数の 2 次元配列 (正方格子グラフに対応)
q = gen_symbols(BinaryPoly, N, N)
# 制約 (w_a で調整)
constraint_x = sum([greater_equal(q[i][j] + q[i + 1][j], 1) for i in range(N - 1) for j in range(N)])
constraint_y = sum([greater_equal(q[i][j] + q[i][j + 1], 1) for i in range(N) for j in range(N - 1)])
constraint = constraint_x + constraint_y
# コスト関数 (w_b で調整)
cost_function = sum([sum_poly(q[i]) for i in range(N)])
# 最終的なエネルギー関数
energy_function = w_a * constraint + w_b * cost_function
```

図 4 Amplify によるエネルギー関数の実装例

図 4 の処理の流れを以下に述べる。

- ① `gen_symbols` メソッドにより、バイナリ変数のサイズ N の 2 次元リストを作成する。このバイナリ変数は正方格子グラフの頂点に対応する。
- ② `greater_equal` メソッドにより、枝の両端のどちらかの頂点が頂点被覆集合に属する制約を課する。頂点被覆集合に属する場合のバイナリ変数の値を 1、属さない場合を 0 とし、枝の両端に対応するバイナリ変数同士を足した際に 1 以上の値になるようにしている。
- ③ `sum_poly` メソッドにより、バイナリ変数の和を最適化項とすることで、頂点被覆集合が最小になるように最適化する。
- ④ 式 (2) の A, B に対応する重み w_a, w_b を、制約項と最適化項に掛け合わせて両者の和を取り、エネルギー関数とする。

3.3 パラメータ設定

本報告で利用した、Amplify の Hitachi クライアントクラスを用いて設定した CMOS アニーリングマシンのパラメータ一覧を表 1 に示す。なお、CMOS アニーリングマシンの情報は[5]で参照できる。

表 1 CMOS アニーリングマシンのパラメータ

HitachClientParameters の属性	属性の 概要	設定値
num_executions	アニーリング実行回数	1
outputs	パラメータ出力の制御	spins=True energies=False execution_time=False
temperature_num_steps	温度変化のステップ数	10
temperature_step_length	温度ステップの長さ	100
temperature_initial	温度の初期値	100.0
temperature_target	最終温度	0.02

なお、温度に関する設定はデフォルト値とし、本実験ではこれらのパラメータは固定している。

4. 予備評価

4.1 実験環境

予備評価の実験環境を表 2 に示す。

表 2 実験環境

利用したマシン及び ライブラリ	詳細
日立 CMOS アニーリングマシン	<ul style="list-style-type: none"> ● Annealing Cloud Web[5] API v2 を使用 ● マシンタイプは GPU 版 (32bit/float)
MacBookAir (macOS Big Sur)	<ul style="list-style-type: none"> ● Python スクリプト (Version 3.8.2) 実行用のマシン ● 1.6GHz Dual Core Intel Core i5 ● メモリ 8GB
Amplify	<ul style="list-style-type: none"> ● アニーリングマシン利用のためのライブラリ ● Version 0.5.13

実験に際して python スクリプト上で制御したパラメータ一覧を表 3 に示す。

表 3 実験で制御したパラメータ

パラメータ	設定値
正方形グラフの 1 辺の長さ N	2~16 (2 刻みで変更)
重み w_a, w_b	0.1~1.0 (0.1 刻みで変更)
アニーリング実行回数 (式 (1) の計算回数)	100

4.2 実験結果

4.2.1 解の精度

正方形の長さ N について、 $N=2$ の問題の最適解回答率の計測結果を図 5 に示す。 $N=4$ の問題の最適解回答率の計測結果を図 6 に示す。

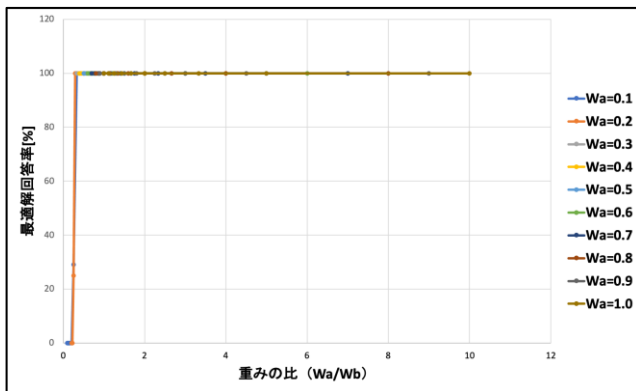


図 5 $N=2$ における重みの比の変更による最適解回答率の推移

図 5 の $N=2$ の問題では、制約項に係る重み w_a が小さい ($w_a \leq 0.2$) 場合、重みの比=0.25 のときに最適解回答率が約 30 [%] となった。また、重みの比 ≥ 0.28 のときに最適解回答率が 100 [%] で安定した。また、制約項に係る重みが大きい ($w_a \geq 0.3$) 場合も、最適解回答率 100 [%] で安定した。

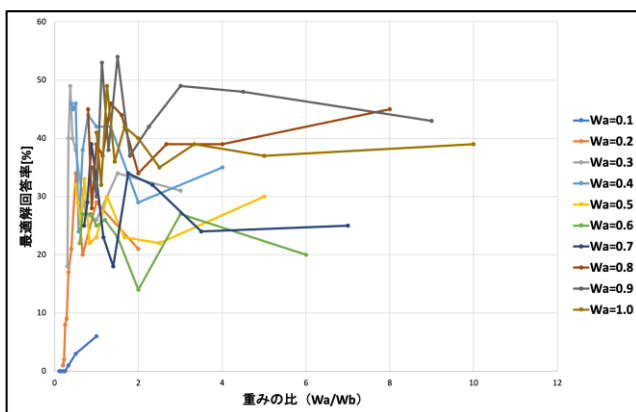


図 6 $N=4$ における重みの比の変更による最適解回答率の推移

図 6 の $N=4$ の問題では、最適解回答率が 100 [%] に到達せず、最大でも 54 [%] となった。各 w_a について、最適解回答率が最も大きくなった重みの比 w_a/w_b を調べると、0.375 ~ 1.5 の範囲であることがわかった。

4.2.2 実行時間

次に、最小頂点被覆問題を解くためのアニーリング時間と、API 利用時の通信時間の両方を含む実行時間の計測結果を図 8 ~ 図 10 に示す。なお、実行時間は図 7 に示す通り time モジュールの perf_counter メソッドにより計測した。

図 7 中の solve メソッドは、アニーリングマシンの実行を行い、最終スピン状態を含む結果を返す。このメソッドが内部で API を呼び出しており、マシン実行のための情報を送信する時間と結果を受信する時間の和が API 利用時の通信時間となる。

```
# 通信時間を含む実行時間の計測
start_time = time.perf_counter()
result = solver.solve(energy_function)
end_time = time.perf_counter()
```

図 7 実行時間計測の実装

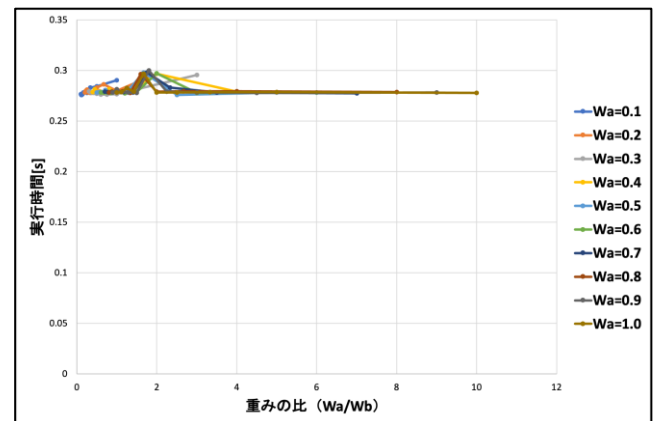


図 8 $N=2$ における重みの比の変更による実行時間の推移

図 8 の $N=2$ の問題では、重みの比の変化による実行時間に大きな変化は見られず、その範囲は 0.276 ~ 0.300 [s] となった。

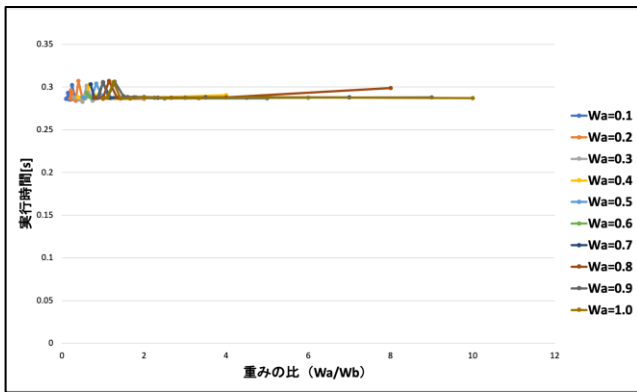


図 9 N=4 における重みの比の変更による実行時間の推移

図 9 の $N=4$ の問題でも、重みの比の変化による実行時間に大きな変化は見られず、その範囲は $0.283 \sim 0.307[s]$ となった。

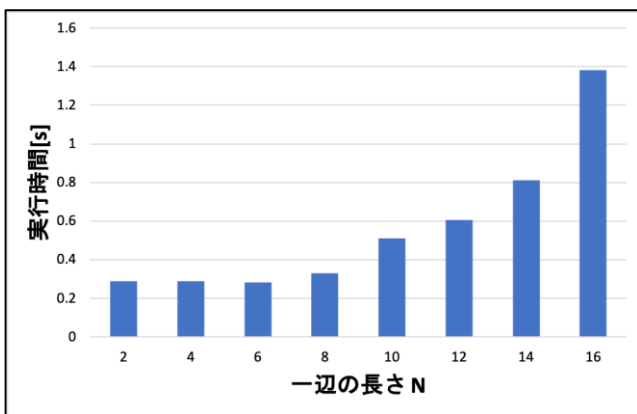


図 10 問題サイズの変更による実行時間の推移
 $((w_a, w_b) = (0.1, 0.1))$ の実行時間

重みの比の変化による実行時間の変化は見られないが、図 10 より、問題サイズによって実行時間は大きく増加することがわかる。最小頂点被覆問題の一辺の長さ N の増加によってグラフの総頂点数は N の 2 乗で増えることから、この結果は妥当だと言える。

一方、 $N=2 \sim 6$ で実行時間が横ばいとなったのは、アニーリング時間に対して通信時間の割合が大きかったことが原因であると思われる。

4.3 考察

4.3.1 重みと解の精度について

本実験では、制約項の重み w_a と最適化項の重み w_b の比を横軸に取り、最適解回答率と実行時間との関係を確認した。

図 6 から、エネルギー関数中の最適化項と比較して制約項の値が大きい場合、つまり X 軸の正の方向に進むにつれ、最適解を導く確率が下がるといえる。しかし、図 5 に示すように、問題サイズが小さく容易に最適解が導ける場合はこの限りではないため、問題サイズに依存するといえる。

また、 $N=4$ において最適解回答率が最も高い $54[\%]$ を記録したときの重みは $(w_a, w_b) = (0.9, 0.6)$, $w_a/w_b = 1.5$ であった。

アニーリングマシンで問題を解くにあたって、制約条件が破綻すると、準最適解を得るところか、そもそも意味のある解が得られないことになる。そのため、最適化項の重みより制約項の重みを少し大きくする、すなわち $w_a/w_b > 1.0$ とするのが妥当であると考えられる。このことから考えると、 $w_a/w_b = 1.5$ のときに最適解回答率が最も高くなったという結果は、この考えと合致する。

4.3.2 アニーリングマシンの実行時間について

図 8～図 10 が示すように、アニーリングマシンの実行時間は、問題サイズに依存するといえる。ただし、 $N=2$ では、 $w_a/w_b = 1.8$ 付近、 $N=4$ では、 $w_a/w_b = 0.25 \sim 1.25$ において実行時間が増加する箇所が見られる。このことから、重みの値もアニーリング時間に影響を及ぼす可能性がある。詳細な分析は今後の課題である。

4.3.3 準最適解について

本実験では、解の精度について最適解を得られる点で評価を行った。一方、CMOS アニーリングマシンの原理を考えると、最適解を得るのではなく、準最適解を得られる精度や、十分実用となる準最適解を得られるまでの実行時間で、性能を評価すべきといえる。本実験では最適解がわかる問題を採用しているため、最適解からの距離を定義することで、準最適解の精度を定義できる。この準最適解における性能評価は、今後の課題である。

5. 関連研究

5.1 量子アニーリングマシンにおける大規模問題の求解

量子アニーリングマシンにおける組合せ最適化問題については、原理的には古典計算機によるアプローチでは実行できない組合せ最適化問題の解を求める事ができる。しかし、量子アニーリングのハードウェアやグラフの接続性の制約によって実用的な問題をマッピングすることができない例が多数ある。そこで Quadratic Unconstrained Binary Optimization (QUBO) の入力インスタンスを適切に分割し、部分問題にするアプローチに適用することで、大規模問題の解を求めることができる可能性がある[6]。

5.2 ソフトウェア自動チューニングへの適用

ソフトウェア自動チューニング (AT) では、問題が分類問題に帰着されることを説明した。ここで分類問題においては、線形分別問題、線形回帰問題、その他の回帰問題、に帰着させることができる。そのため、これらの問題の解法として、線形回帰 (回帰問題用)、ランダムフォレスト (分類問

題用, 回帰問題用), サポートベクターマシン(SVM) (分類問題用) などの既知のアルゴリズムが適用できる.

本報告の冒頭で説明した通り, AT に用いるデータは実行時間など実測されるプログラムを実行したデータであり, 容易に大規模化する. そのため, AT のための分類問題の求解時間の増加が危惧される.

量子アニーリングマシンでATにおける分類問題の求解時間を, 古典的なアルゴリズムの実行より高速に解を求めることができれば, AT 分野に量子アニーリングマシンの普及が期待できる.

5.3 量子アニーリングマシンでの SVM の実行

量子アニーリングマシンにおいて SVM を実行した例では, D-Wave 2000Q での事例[7]がある. 論文[7]によると, 限定された学習データしかないときに, 量子アニーリングマシンの特性による異なる解のアンサンブルから, 解の品質が高いことが報告されている.

一方, 量子インスパイア型イジングマシンでの SVM の適用事例については, イジングモデルとして FPGA 実装した論文[8]など事例があるものの, 著者が知る限り, 現在, ほとんど事例が確認できない.

CMOS アニーリングマシンでの SVM の実装評価は, 重要な今後の課題といえる.

6. おわりに

本研究では, Amplify を利用して日立 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を解き, エネルギー関数中の重みを変更したときの解の変化を評価した.

予備実験の結果, 正方格子グラフの一辺の長さが $N=2$ の場合, 制約項の重みが小さいとき ($w_a \leq 0.2$) に重みの比が $w_a/w_b \geq 0.28$ で最適解回答率が 100 [%]となり, 制約項の重みが大きいとき ($w_a \geq 0.3$) は常に最適解回答率が 100 [%]で安定することが明らかとなった. $N=4$ の場合, 重みの比 $w_a/w_b = 0.375 \sim 1.5$ の範囲で最適解回答率が高く, 最大で 54 [%]となることが明らかとなった.

以上予備実験により, エネルギー関数における, 制約項と最適化項の重みは, (1) 扱う問題サイズに依存すること, (2) 重みの比 w_a/w_b を単に大きくすればよいわけではなく, 適切な比率があることが明らかとなった.

今回取り扱った最適化問題は, 最小頂点被覆問題のみであるが, クラスタリングなどの組合せ最適化問題でも最適化項と制約項の 2 つが表れ, それらの係数値を決定する必要があるため同様の実装や係数値に関する考え方が適用できる.

この重みの比率は, アニーリングマシンのハイパーパラメータと考えられるが, 解の精度のみならず, 誤った値を設

定すると解が破綻するため, 慎重に設定すべきものである.

同時に, 適切な重みの設定のためには試行錯誤が必要であり, 工数を要する. そのため, ソフトウェア自動チューニング (AT) 技術の適用が期待される.

今後は, 準最適解を許容するような評価式の変更や, 初期温度などのアニーリングマシンのハイパーパラメータの変更も追加して, 更なる調査を行う予定である.

また, 大規模問題へのアニーリングマシンの適用は重要な課題である. 特に, 古典アルゴリズムの解の精度と同等な解を得られる場合の, CMOS アニーリングマシンの実行時間について評価する必要がある.

謝辞 CMOS アニーリングマシンの知見, および機材の提供に関し, 日立製作所の山岡雅直氏, および, 小埜和夫氏に感謝の意を表す.

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] T. Katagiri and D. Takahashi, "Japanese Autotuning Research: Autotuning Languages and FFT", Proc. of the IEEE, Vol. 106, Issue 1, pp. 2056-2067, 2018.
- [3] 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, Fundamental Review, Vol.11, No.3, pp.164-171, 2018.
- [4] 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/>
- [5] CMOS アニーリングマシン - Annealing Cloud Web <https://annealing-cloud.com>
- [6] 大山基樹, 森下誠, 片桐孝洋, 大島聡史, 永井 亨, "量子アニーリングマシンにおける組み合わせ最適化問題の適用可能性の調査", 情報処理学会第 83 回全国大会, pp.1-35 -1-36, 2021.
- [7] D.Willsch, M.Willsch, H. De Raedt, K. Michielsen, "Support vector machines on the D-Wave quantum annealer", Computer Physics Communications, Vol. 248, 2020.
- [8] 田口雄大, 飯村凌馬, 河原尊之, "サポートベクターマシンのイジングモデル実装検討", 電子情報通信学会 2021 年総合大会, 2021.