

量子アニーリングと組合せ最適化

倉又 迪哉, 中田 和秀

量子アニーリングは量子効果を用いて組合せ最適化問題を解くアルゴリズムである。近年では D-Wave Systems 社が開発した量子アニーラ（量子アニーリング方式の量子コンピュータ）を利用して、実社会の問題を解こうという試みが見られる。本稿では、量子アニーリングとは何かという点から始め、量子アニーラで組合せ最適化問題を解く手順、実際に組合せ最適化問題を解いたときの性能などについて説明する。

キーワード：量子アニーリング, 量子アニーラ, 組合せ最適化

1. はじめに

量子コンピュータは汎用的な計算が可能なゲート方式と組合せ最適化に特化したアニーリング方式に大別できる。その中でも、本稿ではアニーリング方式に焦点を当てる。量子アニーリングは、西森・門脇が 1998 年に発表した理論 [1] に端を発する。それから 13 年後の 2011 年にカナダの D-Wave Systems 社が世界初の量子アニーラ（量子アニーリング方式の量子コンピュータ）D-Wave One を発表し、大きな注目を集めた。2011 年の発表当初は 128 量子ビットのみであったが年々量子ビット数を伸ばし（図 1）、現在では 5,760 量子ビットの量子アニーラ D-Wave Advantage を利用することができる。これらの技術発展に伴い、近年では実社会で現れる問題の解決を目指す動きも出ている。

本稿では、量子アニーリング [1, 3] とは何かという点から始め、量子アニーラを用いて組合せ最適化問題を解く手順、実際に組合せ最適化問題を量子アニーラで解いたときの性能、量子アニーリングの現状と可能性などについて説明する。

2. 量子アニーリングと QUBO

2.1 量子アニーリングの原理

まず、量子アニーリングの原理について簡単に説明する。量子力学に関連する話に興味のない方は 2.2 節から読んでいただいで問題ない。

量子アニーリングは、量子効果を用いて組合せ最適化問題を解くアルゴリズムである。アルゴリズムの直感的な説明としては、最初に状態の重ね合わせを作り、その程度を弱めることによって最適解を得るものである。準備のため、以下の記号を定義する。

$$\hat{\sigma}^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (1)$$

$$Z_i = \hat{I} \otimes \cdots \otimes \underbrace{\hat{\sigma}^z}_{i \text{ 番目}} \otimes \hat{I} \cdots \otimes \hat{I} \quad (2)$$

$$X_i = \hat{I} \otimes \cdots \otimes \underbrace{\hat{\sigma}^x}_{i \text{ 番目}} \otimes \hat{I} \cdots \otimes \hat{I} \quad (3)$$

ただし、 $\hat{I} \in \mathbb{R}^{2 \times 2}$ は単位行列で、 \otimes はテンソル積である。量子アニーリングで用いられる横磁場イジングモデルのハミルトニアンは式 (4) のように表される。ハミルトニアンとはエネルギーに関する物理量であり、量子力学においては行列（または演算子）の形式をとる。

$$\mathcal{H}(t) = -\sum_{i < j} J_{ij} Z_i Z_j - \sum_i h_i Z_i - \Gamma(t) \sum_i X_i \quad (4)$$

ただし、 $J_{ij} \in \mathbb{R}$ は i 番目の量子ビットと j 番目の量子ビットの相互作用、 $h_i \in \mathbb{R}$ は i 番目の量子ビットにかかる局所磁場を表す。 $\Gamma(t)$ は $\Gamma(0) = +\infty$ 、 $\Gamma(T) = 0$ ($T \gg 0$) となる単調な関数とする。式 (4) は、組合せ最適化問題に関する \mathcal{H}_1 と、状態の重ね合わせに関する $\mathcal{H}_0(t)$ に分けることができる。

$$\mathcal{H}_1 = -\sum_{i < j} J_{ij} Z_i Z_j - \sum_i h_i Z_i \quad (5)$$

$$\mathcal{H}_0(t) = -\Gamma(t) \sum_i X_i \quad (6)$$

量子アニーリングにおけるハミルトニアン $\mathcal{H}(t)$ の挙動を時刻ごとに説明すると次のようになる。まず $t = 0$ において、 $\mathcal{H}(0) \simeq \mathcal{H}_0(0)$ である。時刻 t が進むにつれて $\Gamma(t)$ が小さくなるため、相対的に \mathcal{H}_1 が大きくなる。そして、終時間 $t = T$ においては $\mathcal{H}(T) \simeq \mathcal{H}_1$ となる。このハミルトニアンの変化に対して、状態は次のように変化する。まず $t = 0$ において $\mathcal{H}_0(0)$ の基底

くらまた みちや, なかた かずひで
東京工業大学工学院経営工学系
〒 152-8552 東京都目黒区大岡山 2-12-1

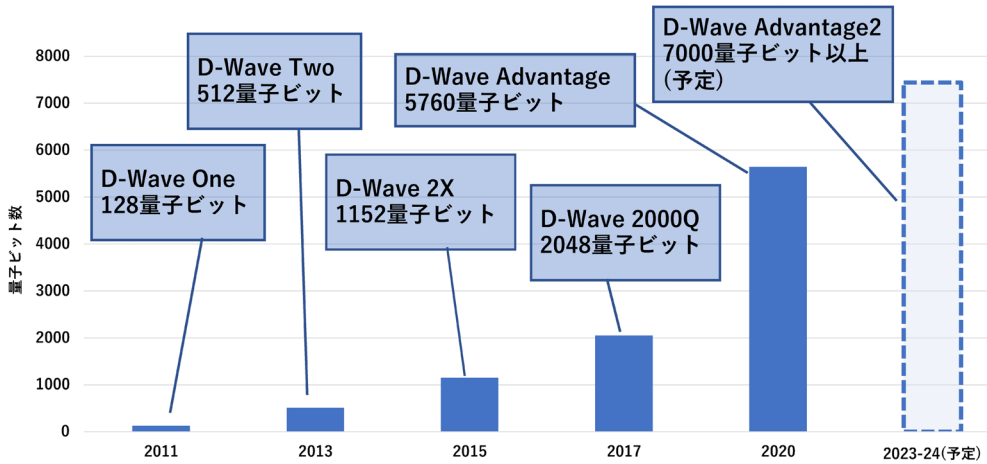


図1 D-Wave Systems の量子アニーラの量子ビット数 [2]

状態からスタートし、十分ゆっくり遷移させ¹、終時間 $t = T$ において \mathcal{H}_1 の基底状態を得る。この基底状態が組合せ最適化問題の最適解に相当する。つまり、量子アニーリングでは四則演算や論理操作は一切行わず、最初にハードウェア上で状況設定を行うと、後は量子力学の物理現象を利用して最適解を探す。なお、厳密解を得ることに対する統計的な解析はあるものの、実際に量子アニーラを用いて組合せ最適化問題を解く場合は近似解法と見るのが適切である。

2.2 量子アニーラで組合せ最適化問題を解く

2.1 節では、量子アニーリングは \mathcal{H}_1 の基底状態を求めていると述べた。この \mathcal{H}_1 は、イジングモデルと呼ばれる物理モデルである (ちなみに \mathcal{H}_0 は横磁場項である)。2.1 節ではイジングモデル \mathcal{H}_1 を量子力学的に記述したが、2.2 節では古典的に記述する。このとき、イジングモデルはエネルギーが

$$E(\mathbf{s}) = -\sum_{i < j} J_{ij} s_i s_j - \sum_i h_i s_i \quad (\forall i, s_i \in \{-1, 1\}) \quad (7)$$

で与えられる磁性体のモデルとなる ($J_{ij}, h_i \in \mathbb{R}$)。イジングモデルでは各スピン変数 $s_i \in \{-1, 1\}$ が上向きか下向きかを向く微小な磁石を表しており、スピン変数同士が相互作用を通じて互いに影響を及ぼし合っている。量子アニーリングでは、このように定義されたエネルギーが最小となるスピン \mathbf{s} を探し出す。式 (7) は、式 (5) の Z_i をスピン変数 s_i に置き換えたものであることが確認できる (なぜ置き換えたものになるかは、Appendix で説明を行った)。

¹ 量子の世界での「ゆっくり」であり、実際の量子アニーラでは 50 マイクロ秒程と短時間である。

式 (7) に対して、 $q_i = (s_i + 1)/2$ として変数変換を行うと、 q_i は 0 か 1 を取る変数になる。さらに $q_i^2 = q_i$ であることを利用すると、式 (7) の最小化と等価な最適化問題 (8) を構築することができる。

$$\begin{aligned} \min \quad & \mathbf{q}^\top Q \mathbf{q} \\ \text{s.t.} \quad & \mathbf{q} \in \{0, 1\}^L. \end{aligned} \quad (8)$$

ただし、 L は 2 値変数の個数で、 Q は変数変換の結果得られた $L \times L$ の実対称行列である。この問題は Quadratic Unconstrained Binary Optimization (QUBO) と呼ばれ、量子アニーリングを利用するときの標準形である。解きたい組合せ最適化問題を QUBO として定式化すれば、先程と逆の変換をたどり等価なイジングモデルを得ることができる。そしてそのイジングモデルのエネルギー最小点を量子アニーラが見つかる。

実際に D-Wave Systems の量子アニーラを使って組合せ最適化問題を解いてみたい方は、*D-Wave Leap*² に登録することをおすすめする。開発したコードをオープンソースとすることで、毎月一定時間無料で利用できる。また、実際に開発をする際には、文献 [4] が大変参考になる。

2.3 量子アニーラのハードウェア

通常の計算で用いる古典コンピュータ³は Central Processing Unit (CPU) で計算を実行するが、量子アニーラは Quantum Processing Unit (QPU) で計算を実行する。QPU には量子ビットが配置されており、

² <https://cloud.dwavesys.com/leap/>

³ 現在一般的に使われているコンピュータのことである。量子コンピュータと区別するために呼んでいるのであって古いという意味はない。

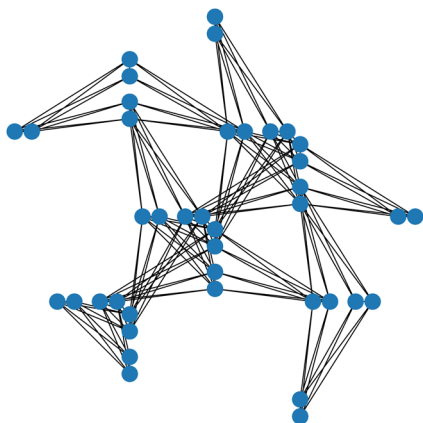


図2 ベガサスグラフの部分構造
丸が量子ビット、実線が量子ビット間の結合を表す。

その配置の仕方を表現するためにグラフが用いられる。このグラフでは、量子ビットをノード、量子ビット間の接続をエッジとする。図2はD-Wave Advantageにおける量子ビットの配置を表しており、ベガサスグラフと呼ばれる。見てわかるように量子ビット間に必ずしも接続があるわけではなく、スパースな構造である。

2.4 量子アニーラ上でのQUBOの表現

2.2節で説明したイジングモデルは、そのスピン変数を量子アニーラにおける量子ビットに割り当てる作業が必要である。この割り当てを「埋め込み」という。2.3節で説明したように、量子アニーラは量子ビット間の接続が疎であるため、任意のスピン間の相互作用を表現することができない。したがって、多くの場合そのままイジングモデルを量子アニーラのQPU上で構築することができないことになる。そのため、イジングモデルの一つのスピン変数すなわちQUBOの一つの論理変数に対し、複数の量子ビットを対応させるような埋め込みを考える必要がある。図3は、そのような埋め込みのイメージである。この例は、密な論理構造をもつQUBO(最左図)を、ベガサスグラフに見立てた疎なグラフ(中央図)上で表現する埋め込み(最右図)を表している。最右図において、二つの量子ビット1,2を用いて論理変数 x_a を構成している。このように冗長な量子ビットを導入することで密なQUBOを量子アニーラで表現できるようになる。その一方で量子ビットを余分に消費することになる。このため、たとえばQUBOの係数行列 Q が完全に密行列となる場合、5,760量子ビット搭載のD-Wave Advantageに埋め込むと、利用できるQUBOの論理変数の数は180程度に制限されてしまう。

あるQUBOがあったとき、使用する量子ビット数

が最小となる埋め込みを得ることは難しい(これ自体が組合せ最適化問題である)。Cai et al. [5]は、古典コンピュータ上で動作する埋め込み作成アルゴリズムを提案した。これは、ダイクストラ法をサブルーチンにもつヒューリスティック解法である。この実装は、D-Wave Systemsが開発するPythonパッケージ群⁴に含まれており、多くのユーザーに利用されている。現在、量子ビット数が少なくなる埋め込みを高速に求めるアルゴリズムの開発は重要な問題であり、その実装には大きなニーズがある。

最後に、ユーザーが量子アニーラを用いて組合せ最適化問題を解くワークフローを示す。

1. 解きたい組合せ最適化問題をQUBOで記述する⁵
 2. QUBOをイジングモデルに変換する
 3. イジングモデルを量子アニーラに埋め込む
 4. 量子アニーラで、埋め込まれたイジングモデルを解いて、その解を取得する
 5. 埋め込まれたイジングモデルの解から、解きたい組合せ最適化問題の解を復元する
1. は、自分で定式化をする必要がある。3. は、埋め込みアルゴリズムとしてCai et al. [5]のアルゴリズムを用いる場合はパッケージがやってくれる。それ以外はパッケージが自動で処理を行う。まとめると、ユーザーは1.にのみ注力すればよいことになる。ただし、より良い解を得るためには、1.の定式化を工夫すること、4.における量子アニーラのパラメータを適切に設定することが必要になる。

3. 線形等式制約の扱い方

3.1 ペナルティ関数法

3節では、線形等式制約を含む組合せ最適化問題について考える。具体的には、次の形式の最適化問題を対象とする。

$$\begin{aligned} \min \quad & f_0(\mathbf{q}) \\ \text{s.t.} \quad & f_k(\mathbf{q}) = C_k, \quad \forall k \in \{1, 2, \dots, M\}, \quad (9) \\ & \mathbf{q} \in \{0, 1\}^L. \end{aligned}$$

ただし、 $f_0(\mathbf{q})$ は \mathbf{q} の2次以下の関数、 $f_k(\mathbf{q})$ は \mathbf{q} の1次の関数とする。QUBO(8)とは形式が異なるが、線形等式制約 $f_k(\mathbf{q}) = C_k$ が加わっているところが本質的な違いとなる。

⁴ “dwave-ocean-sdk”. 以下のリンクから入手可能。

<https://github.com/dwavesystems/dwave-ocean-sdk>

⁵ 記述にはPyQUBOなどのパッケージが利用できる。

<https://github.com/recruit-communications/pyqubo>

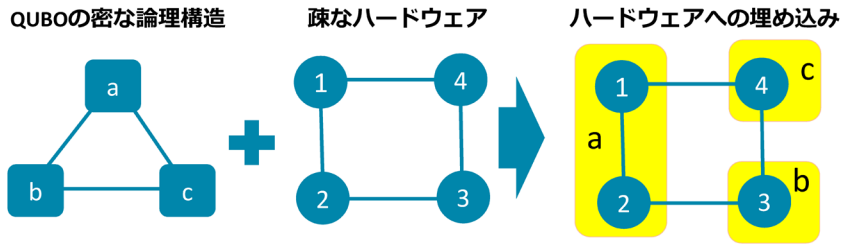


図3 埋め込み
 最左図は QUBO の論理構造を表している。ここで、 a, b, c は論理変数 $x_a, x_b, x_c \in \{0, 1\}$ を表すとする。三つの論理変数の間にはエッジがあるので、QUBO の目的関数は、 $x_a x_b$ と $x_b x_c$ と $x_c x_a$ の項を含む。中央図は、量子アニーラの QPU におけるグラフを表す。1, 2, 3, 4 はそれぞれ量子ビットである。最右図では、論理変数 a を量子ビット 1, 2 が表している。また、論理変数 b を量子ビット 3、論理変数 c は量子ビット 4 が表している。

量子アニーラではバイナリ制約以外の制約を含んだ問題を解くことはできない。このため、制約違反をペナルティとして目的関数に加えるペナルティ関数法がよく使われる。つまり、最適化問題 (10) を解くことになる。

$$\begin{aligned} \min \quad & f_0(\mathbf{q}) + \frac{1}{2} \sum_{k=1}^M \lambda_k (f_k(\mathbf{q}) - C_k)^2 \\ \text{s.t.} \quad & \mathbf{q} \in \{0, 1\}^L. \end{aligned} \quad (10)$$

ここで、 $\lambda_k (k = 1, 2, \dots, M)$ は正のペナルティパラメータである。ただし、この問題の目的関数に含まれる $\frac{1}{2} \sum_{k=1}^M \lambda_k (f_k(\mathbf{q}) - C_k)^2$ は、QUBO で表現したときの係数行列 Q を密にする可能性がある。すなわち、埋め込みの際に冗長な量子ビットを数多く使用しなくてはならなくなる。そのため、たとえば 5,760 量子ビットを使っても、2 次割当問題や巡回セールスマン問題はサイズが 13 程度の問題しか埋め込めない。また、ペナルティパラメータの設定も難しく⁶、実際に計算してもサイズ 9 程度から実行可能解を得ることが困難であった。このように、バイナリ制約以外の制約をもった組合せ最適化問題を量子アニーラで解くことは難易度が上がる。

3.2 Ohzeki [6] のアルゴリズム

Ohzeki [6] は、量子アニーラから得られる解の統計的な性質を利用して、線形等式制約を含んだ組合せ最適化問題を解く全く新しいアルゴリズムを提案した⁷。この方法は、結果として最適化問題 (9) のラグランジュ双対問題

⁶ 量子アニーラ上で物理的に環境を構築するため、古典コンピュータほど細かく定数の設定ができない。

⁷ 大関先生は Youtube チャンネルで講義を公開しており勉強になるうえに面白い。 https://www.youtube.com/channel/UCxgJJrnet1pNM_JJA--Y6wx

$$\max_{\lambda} \min_{\mathbf{q} \in \{0, 1\}^L} f_0(\mathbf{q}) - \sum_{k=1}^M \lambda_k (f_k(\mathbf{q}) - C_k) \quad (11)$$

に対し、劣勾配法を適用したものと近いものになっている。本節では、物理的な用語を極力排して、このアルゴリズムについて解説を行う。

このアルゴリズムは、量子アニーラから得られた解が Gibbs-Boltzmann 分布に従うことを利用する。まず準備として、この分布について説明する。Gibbs-Boltzmann 分布では、スピン変数 $\mathbf{s} \in \{-1, 1\}^L$ に対する確率 $p(\mathbf{s})$ は式 (12) のように表される。

$$p(\mathbf{s}) = \frac{1}{Z} \exp(-\beta E(\mathbf{s})) \quad (12)$$

ただし、 $\beta > 0$ は逆温度と呼ばれるパラメータである。また、 $E(\mathbf{s})$ は式 (7) と同一である。 Z は確率の総和を 1 に正規化するための規格化定数であり、分配関数と呼ばれる。分配関数は式 (13) で定義される。

$$Z = \sum_{\mathbf{s}} \exp(-\beta E(\mathbf{s})) \quad (13)$$

ここで、 $\sum_{\mathbf{s}}$ は、すべての $\mathbf{s} \in \{-1, 1\}^L$ に対して足し合わせるという意味である。

文献 [6] では、Hubbard-Stratonovich transformation [7, 8] を用いた導出と、KL ダイバージェンスを用いた導出が説明されている。本稿では後者の導出について最適化の視点から説明する。

目的関数 $f_0(\mathbf{q})$ のみを組み入れた量子アニーラから得られた解は、先程説明したように Gibbs-Boltzmann 分布に従う。この分布を P としたとき、 $\mathbf{q} \in \{0, 1\}^L$ となる確率 $P(\mathbf{q})$ は、次のように表せる。

$$P(\mathbf{q}) = \frac{1}{Z} \exp(-\beta f_0(\mathbf{q})) \quad (14)$$

ただし、 Z は適切に定められた分配関数である。この

分布は「 $f_0(\mathbf{q})$ の値が小さいとき、その確率 $P(\mathbf{q})$ が大きくなる」という性質をもっている。これは、サンプリングによって $f_0(\mathbf{q})$ の最小解が得られる可能性が高い、という都合のよいものである。しかし、制約条件は一切考慮されていない。そこで、分布 P の代わりに、確率的に制約条件を満たし、かつ分布 P に近い分布 Q を利用することを考える。こうすることで、ある程度制約条件を満たしながら、 $f_0(\mathbf{q})$ の値が小さな解を見つけることが可能となる。まず、 Q は分布であることから、以下の条件を満たしている必要がある。

$$\sum_{\mathbf{q}} Q(\mathbf{q}) = 1, \quad (15)$$

$$Q(\mathbf{q}) \geq 0, \quad \forall \mathbf{q} \in \{0, 1\}^L. \quad (16)$$

そして、制約条件は分布 Q に対して期待値で満たす、という次の条件を課す。

$$\langle f_k(\mathbf{q}) \rangle_Q = C_k, \quad \forall k \in \{1, 2, \dots, M\} \quad (17)$$

ここで、 $\langle f_k(\mathbf{q}) \rangle_Q$ は分布 Q に対する $f_k(\mathbf{q})$ の期待値 $\langle f_k(\mathbf{q}) \rangle_Q = \sum_{\mathbf{q}} Q(\mathbf{q}) f_k(\mathbf{q})$ である。次に、分布 P に近い分布 Q を見つけるにあたって、分布 P と Q の近さを考える。そのとき、次で定義されるKLダイバージェンスを利用する。

$$KL(Q|P) = \sum_{\mathbf{q}} Q(\mathbf{q}) \ln \left(\frac{Q(\mathbf{q})}{P(\mathbf{q})} \right) \quad (18)$$

以上の議論より、分布 Q は(15), (16), (17)を満たしつつ、分布 P とのKLダイバージェンス(18)が最小となるものがよい。これは $Q(\mathbf{q})$ ($\mathbf{q} \in \{0, 1\}^L$)を変数とする、次の最適化問題として記述できる。

$$\begin{aligned} \min \quad & KL(Q|P) \\ \text{s.t.} \quad & \sum_{\mathbf{q}} Q(\mathbf{q}) = 1, \\ & Q(\mathbf{q}) \geq 0, \quad \forall \mathbf{q} \in \{0, 1\}^L, \\ & \langle f_k(\mathbf{q}) \rangle_Q = C_k, \quad \forall k \in \{1, 2, \dots, M\}. \end{aligned} \quad (19)$$

この問題は凸計画問題である。そして、Lagrange 双対は次の最適化問題となる。

$$\begin{aligned} \min \quad & - \sum_{k=1}^M C_k v_k - \frac{1}{\beta} \log Z \\ \text{s.t.} \quad & f_{\text{new}}(\mathbf{q}) = f_0(\mathbf{q}) - \sum_{k=1}^M v_k f_k(\mathbf{q}), \\ & Z = \sum_{\mathbf{q}} \exp(-\beta f_{\text{new}}(\mathbf{q})), \\ & Q(\mathbf{q}) = \frac{1}{Z} \exp(-\beta f_{\text{new}}(\mathbf{q})) \quad \forall \mathbf{q}. \end{aligned} \quad (20)$$

制約条件は $\mathbf{v} \in \mathbb{R}^M$ が決まれば順に計算していけるものなので、これは実質的に \mathbf{v} に関する無制約の凸最適化問題とみなせる。また、最適解 \mathbf{v}^* から順に計算さ

れた $Q(\mathbf{q})$ は、われわれが求めたかった主問題(19)の最適解となる。双対問題(20)の目的関数を $g(\mathbf{v})$ とすると、その勾配(偏微分)は

$$\frac{\partial g}{\partial v_k} = -C_k + \langle f_k(\mathbf{q}) \rangle_Q \quad (k = 1, 2, \dots, M) \quad (21)$$

と計算できる。このため、双対問題に最急降下法を適用した場合、 \mathbf{v} の更新式は式(22)となる。

$$v_k^{(t+1)} \leftarrow v_k^{(t)} + \eta \left(C_k - \langle f_k(\mathbf{q}) \rangle_{Q^{(t)}} \right), \quad \forall k \quad (22)$$

$\mathbf{v}^{(t)}$ の t は、アルゴリズムの t イテレーション目を表す。また、 $Q^{(t)}$ は t イテレーション目における Q を表す。 η はステップサイズである。

期待値 $\langle f_k(\mathbf{q}) \rangle_Q$ の計算を古典コンピュータで効率的に行うことは難しい。しかし、分布 Q はその定義から、 $f_{\text{new}}(\mathbf{q})$ を目的関数とした量子アニーラの解が従うGibbs-Boltzmann分布となっている。つまり、この量子アニーラでサンプリングされた解は自然と分布 Q に従うため、期待値計算を効率的に行うことができる。すなわち、量子アニーラによってサンプリングされた解 $(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)$ に対して、以下のように計算することが可能である。

$$\langle f_k(\mathbf{q}) \rangle_Q \approx \frac{1}{n} \sum_{i=1}^n f_k(\mathbf{q}_i), \quad \forall k \quad (23)$$

さらに、この量子アニーリングはペナルティ関数法で行う量子アニーリングのときよりも、良い性質をもっている。新しい目的関数は $f_{\text{new}}(\mathbf{q}) = f_0(\mathbf{q}) - \sum_{k=1}^M v_k f_k(\mathbf{q})$ である。つまり、最適化問題(10)の目的関数に現れる2次のペナルティ項が線形項になっている。そのため、係数行列 Q の疎性は悪化しない。これにより、最適化問題(10)と比べて、より大きな問題を量子アニーラに埋め込めるようになる。

以上をまとめて、文献[6]で提案されたアルゴリズムを次に示す。

Algorithm 1 Ohzeki [6] のアルゴリズム

Ensure: best \mathbf{q}_{best}

1: **Initialize:**

2: $t \leftarrow 0$

3: $\mathbf{v}^{t=0} \leftarrow \mathbf{0}$

4: **while** some condition **do**

5: $(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n) \leftarrow$ Sampling multiple \mathbf{q} from a distribution $Q^{(t)}$ by using a quantum annealer

6: $v_k^{(t+1)} \leftarrow v_k^{(t)} + \eta \left(C_k - \frac{1}{n} \sum_{i=1}^n f_k(\mathbf{q}_i) \right), \quad \forall k \in \{1, \dots, M\}$

7: $t \leftarrow t + 1$

8: **end while**

9: Save $\mathbf{q}_{\text{best}} \leftarrow \operatorname{argmin}_{\mathbf{q} \in \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}} f_0(\mathbf{q})$

5 行目は、分布 $Q^{(t)}$ に従う n 個のサンプル $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ を量子アニーラから得るということの意味する。9 行目はサンプルの中から目的関数値が最も良い解を保存している。Ohzeki [6] によるアルゴリズムは、サンプリングによって期待値を近似している点と、期待値が正確に求まったとしても厳密解へ収束するかはわからないという点で、ヒューリスティックなアルゴリズムとなっている。

最後に、Ohzeki [6] のアルゴリズムに関連した筆者らの研究 [9] を紹介する。Ohzeki [6] のアルゴリズムを 2 次割当問題に適用した場合、実行可能解は得られないが、その制約違反の程度は軽いという傾向があることが判明した。そこで、Ohzeki [6] のアルゴリズムで得られた 2 次割当問題の解に対し、必ず実行可能解となるよう解を微調整するヒューリスティックなアルゴリズムを提案し適用した。その結果、ペナルティ関数法を用いた過去の研究 [10] では、サイズが 8 の 2 次割当問題しか解けなかったのに対し、サイズが 19 までの 2 次割当問題を解くことに成功した⁸。

3.3 不等式制約をもつ場合

量子アニーリングでは不等式制約の扱いは非常に難しいことが知られている。もし、不等式 $f_k(\mathbf{q}) \leq C_k$ の左辺の係数が非負の整数値しか取らない場合は、スラック変数 s を導入して、 $f_k(\mathbf{q}) + s = \lfloor C_k \rfloor$ とする。そして、この s を $\lceil \log_2(C_k + 1) \rceil$ 個の量子ビットで表現すれば、等式制約に変更することができる。ただし、このような操作はさらに問題を複雑にするため、現実的な方法とはなりにくい。

4. 量子アニーラの応用

4.1 量子アニーラの応用分野の概観

量子ビット数の増加や QPU の改良を経て、近年では量子アニーラを用いて実社会における最適化問題を解こうとする野心的な挑戦が続いている。量子アニーラが登場してから、わずか数年のうちに実社会の問題を解けるまでの計算能力を得たのは、まさに長足の進歩である。

しかし、ハードウェアが発展途上であったため仕方のない面もあるが、過去の多くの量子アニーリングを用いた応用研究では、古典コンピュータの優れたアルゴリズムとの比較がされていない。その数少ない例外として、論文 [11, 12] がある。この二つの論文では、混合整数計画ソルバー Gurobi [13] と量子アニーラで解

いた比較結果が載せられており、大変参考になる。しかしながら、実用的な組合せ最適化問題において、量子アニーラが混合整数計画ソルバーを上回るような結果とはなっていない。

実務家が量子アニーラを用いるか検討する際の最大の関心事は、量子アニーラが古典コンピュータよりも優れているか否かである。そのため、技術的に成熟していく量子アニーラのパフォーマンスを確認するために、Gurobi [13] や CPLEX [14], Nuorium Optimizer [15] のような高性能なソルバーと比較することは今後ますます重要になると考える。

4.2 スポーツスケジューリングへの応用

最近の筆者らの研究 [16] では、量子アニーラと混合整数計画ソルバーの精度や計算速度の比較を行っている⁹。ここでは、量子アニーラが「mirrored double round-robin tournament におけるブレイク数最小化問題」を効率よく解くことが、その理由とともに示されている。量子アニーラが混合整数計画ソルバーに勝るパフォーマンスとなる初の実用的な組合せ最適化問題であり、以下で紹介する。

4.2.1 mirrored double round-robin tournament におけるブレイク数最小化問題

スポーツスケジューリングは、スポーツの大会において適切なスケジュールを求めることを指す。この問題は、多くの実務上の要請があり、そこから多様な組合せ最適化問題へと派生している [17, 18]。ここでは、mirrored double round-robin tournament (MDRRT) と呼ばれるトーナメント形式において、ブレイクと呼ばれる好ましくない状況ができるだけ少なくなるように、各試合の試合会場を決定する問題を扱う。

まず、double round-robin tournament (DRRT) を、以下の 1.–4. の性質をもつトーナメントであると定義する。

1. 各チームは自分以外のチームと 2 回ずつ対戦する。
2. 各チームはホームをもつ。
3. 各試合は、対戦する 2 チームどちらかのホームで開催する。
4. 2 回対戦があるうち、最初の試合をホーム（アウェイ）で開催したら、2 試合目はアウェイ（ホーム）で開催する。

MDRRT は、ホームとアウェイを交換することを除いて、前半戦と後半戦の試合の組み合わせが等しい DRRT である。

⁸ 近似解を得たという意味である。

⁹ GitHub においてコード公開をしている。 <https://github.com/MK-tech20/MDRRT-scheduling-dwave>

表 1 チーム数 4 のときの MDRRT のタイムテーブル

Slot	1	2	3	4	5	6
team 1	2	3	4	2	3	4
team 2	1	4	3	1	4	3
team 3	4	1	2	4	1	2
team 4	3	2	1	3	2	1

縦軸はチーム、横軸にはスロット (対戦日) が並んでおり、各セルは対戦相手の番号が格納されている。

表 2 チーム数 4 のときの MDRRT の Home away assignment

Slot	1	2	3	4	5	6
team 1	H	A	H	A	H	A
team 2	A	A	H	H	H	A
team 3	H	H	A	A	A	H
team 4	A	H	A	H	A	H

縦軸はチーム、横軸にはスロット (対戦日) が並んでおり、各セルはホームゲームならば“H”、アウェイゲームならば“A”が入っている。

表 1 は、チーム数 4 のときの MDRRT のタイムテーブルを表している。タイムテーブルとは各試合の対戦相手とそのスロット (試合日) を示す表である。たとえば、team 2 の slot 4 は 1 であるが、これはチーム 2 は 4 スロット目でチーム 1 と対戦することを意味する。また、前半戦 (1, 2, 3 スロット) と後半戦 (4, 5, 6 スロット) は等しい対戦相手の組み合わせをもつことが確認できる。

表 2 は home-away assignment (HA-assignment) を表している。HA-assignment は各試合が対戦相手のどちらのホームで開催するかを示す。縦軸と横軸はタイムテーブルと同様で、各セルはホームなら“H”、アウェイなら“A”となる。

次に、ブレイクについて説明する。ブレイクとは、各チームにとってホームゲーム、またはアウェイゲームが連続することである。一般に、ブレイクの少ないスケジュールが好ましいとされている。表 2 中の“H-H”と“A-A”がブレイクである。たとえば、team 2 の slot2 はアウェイが連続しているため、ブレイクが発生している。表 2 の合計ブレイク数は 6 である。

ここまで、MDRRT、タイムテーブル、HA-assignment について説明してきた。残りは、MDRRT におけるブレイク数最小化問題を定義するのみである。MDRRT におけるブレイク数最小化問題とは、対戦相手の組み合わせ (タイムテーブル) が与

えられたうえで、ブレイク数が最小な開催地の割当 (HA-assignment) を見つける問題のことである。

4.2.2 計算時間の比較

ブレイク数最小化問題に対する古典コンピュータを用いた先行研究として、制約プログラミング [19]、混合整数計画ソルバーを用いる手法 [20, 21]、近似手法 [22] などが挙げられる。同問題を量子アニーラで解くにあたって、文献 [16] では Urdaneta et al. [21] による定式化を利用し QUBO を構築している。

量子アニーラは、量子状態を保持できる時間に物理的な制限があるため、計算時間を伸ばすことができない。言いかえると、量子アニーラでは短い時間で近似解を得ることのみ可能である。このため、量子アニーラが 0.05s の間に得た近似解に対し、その目的関数値に混合整数計画ソルバーが到達するためにかかった時間を計測し比較を行う。量子アニーラの挙動を基準としており、量子アニーラに有利な実験の設定になっていることに注意が必要である。計算環境はそれぞれ次のとおりである。

- 量子アニーリング：量子アニーラは D-Wave Advantage (Advantage-system1.1) を使用した。
- 混合整数計画ソルバー：計算には Gurobi Optimizer (version 9.1.2) [13] を使用した。計算機は Intel(R) Core(TM) i7-7700HQ CPU 2.80GHz, 4 コア, 8 スレッドである。

なお、混合整数計画ソルバーで解く際には、Urdaneta et al. [21] による定式化を利用した。MDRRT におけるブレイク数最小化問題の定式化は幾つか知られているが、筆者の知る限りこの定式化が最も計算効率が良い。

実験では、サイズごとにランダムに 5 問作成し、結果はその平均を用いた。結果は表 3 にまとめている。たとえば、チーム数 36 の問題に対して、量子アニーラが 0.05s で求めた目的関数値にたどり着くまでに、混合整数計画ソルバーは平均 84.8s 必要とすることがわかる。この意味で、量子アニーラで解いたほうが、古典コンピュータ上で混合整数計画ソルバーで解くよりも優れているといえる。ただし、既に述べたように量子状態を保持できる時間には制限があるため、量子アニーラは計算時間を伸ばしてもさらに良い解が得られるというわけではない。一方で、混合整数計画ソルバーを使った場合は計算時間を伸ばせば、より良い解が得られることが期待できるという利点がある。

文献 [16] では、量子アニーラが解きやすい組合せ最適化問題の特徴について考察し、追加的な検証も行っ

表 3 MDRRT におけるブレイク数最小化問題に対して、0.05 s で量子アニーラが求めた目的関数値に、混合整数計画ソルバーが到達するのにかかる時間

チーム数	混合整数計画ソルバー
4	0.039s
8	0.062s
12	0.148s
16	0.352s
20	2.837s
24	8.021s
28	26.003s
32	49.272s
36	84.806s
40	75.338s
44	57.134s
48	6.881s

ている。結論だけを述べると、以下 2 点が今回の量子アニーラの優れたパフォーマンスに繋がったと考えられる。

- QUBO の係数行列 Q がスパースであること
- バイナリ制約以外の制約をもたないこと

5. 理論としての量子アニーリングの広がり

量子アニーリングの理論は、量子ゲートのアルゴリズムにも影響を与えている。近年、ノイズのある量子ゲートを用いて組合せ最適化を解く手法として Quantum Approximate Optimization Algorithm (QAOA) [23] が注目されている。QAOA は、量子アニーリングを離散化して量子ゲートに落とし込んだアルゴリズムともいえる。

また、現状の量子アニーラは、2000 年までに発表された量子アニーリングの最も標準的な理論 [1, 2] をハードウェアに実装したものである。その後、より工夫を凝らした量子アニーリングの理論 [24] も発表されており、今後も理論とハードウェアの両輪の発展が期待される。

6. まとめ

量子アニーリングの研究には、量子力学や統計物理学に関連する要素が色濃い研究もあれば、オペレーションズ・リサーチと強く関連する研究もある。その中でも本稿ではオペレーションズ・リサーチ寄りの内容を扱った。

本文中で述べたように、QUBO の形で定式化された組合せ最適化問題は、量子アニーラを用いると高速に近似解を求めることが可能である（この用途に限定

される）。しかし、現状の量子アニーラによって組合せ最適化問題を解くことは、混合整数計画ソルバーに比べ扱いにくい点が多い。特に、量子ビット間の疎性によって、実質的に使える変数の数が少なくなることで、バイナリ制約以外の制約条件が（多数）ある場合は実行可能解を出力するのに苦労することが課題として挙げられる。これまで、筆者はさまざまなタイプの組合せ最適化問題を量子アニーラで解くことを試みたが、量子アニーラの（混合整数計画ソルバーとの対比）性能は、解く問題に強く依存した。たとえば、MDRRT におけるブレイク数最小化問題に対しては量子アニーラは素晴らしい結果を得たが、巡回セールスマン問題、ナップザック問題などに対しては実用的ではなかった。

しかしながら、従来のアルゴリズムとは全く異なる原理に基づく量子アニーリングの魅力は際立っていると感じている。基礎的な理論やハードウェア、そして応用研究が互いに影響を与え合う当分野の発展を筆者自身楽しみにしている。

謝辞 本稿執筆の機会を頂き、感謝申し上げます。また、株式会社エヌ・ティ・ティ・データの香月諒大氏から本稿について有益なアドバイスを頂きました。感謝致します。また、筆者の二つの研究 [9, 16] は、株式会社エヌ・ティ・ティ・データとの共同研究です。この場を借りて御礼申し上げます。

参考文献

- [1] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” *Physical Review E*, **58**, pp. 5355–5363, 1998.
- [2] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser, “Quantum computation by adiabatic evolution,” *arXiv*, arXiv:quant-ph/0001106, 2000.
- [3] D-Wave Systems, The D-Wave Clarity Roadmap, https://www.dwavesys.com/media/xvjpraig/clarity-roadmap-digital_v2.pdf (2022 年 3 月 12 日閲覧)
- [4] 香月諒大, “NTT データ 量子コンピューティングガイドライン 量子アニーリング/イジングマシン編,” https://www.nttdata.com/jp/ja/news/services_info/2021/012800/ (2022 年 2 月 6 日閲覧)
- [5] J. Cai, W. G. Macready and A. Roy, “A practical heuristic for finding graph minors,” *arXiv*, arXiv:1406.2741, 2014.
- [6] M. Ohzeki, “Breaking limitation of quantum annealer in solving optimization problems under constraints,” *Scientific Reports (Nature Publisher Group)*, **10**, no. 3126, 2020.
- [7] J. Hubbard, “Calculation of partition functions,” *Physical Review Letters*, **3**, pp. 77–78, 1959.
- [8] R. L. Stratonovich, “On a method of calculating quantum distribution functions,” *Soviet Physics Doklady*, **2**, p. 416, 1957.

- [9] M. Kuramata, R. Katsuki and K. Nakata, “Larger sparse quadratic assignment problem optimization using quantum annealing and a bit-flip heuristic algorithm,” In *Proceedings of the 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA)*, pp. 556–565, 2021.
- [10] N. Nishimura, K. Tanahashi, K. Suganuma, M. J. Miyama and M. Ohzeki, “Item listing optimization for e-commerce websites based on diversity,” *Frontiers in Computer Science*, **1**, 2, 2019.
- [11] M. Ohzeki, A. Miki, M. J. Miyama and M. Terabe, “Control of automated guided vehicles without collision by quantum annealer and digital devices,” *Frontiers in Computer Science*, **1**, 9, 2019.
- [12] D. O’Malley, V. V. Vesselinov, B. S. Alexandrov and L. B. Alexandrov “Nonnegative/binarymatrix factorization with a d-wave quantum annealer,” *PLoS ONE*, **13**, e0206653, 2018.
- [13] Gurobi Optimization, Gurobi Optimizer Reference Manual, <https://www.gurobi.com> (2022年2月6日閲覧)
- [14] IBM, IBM ILOG CPLEX Optimizer, <https://www.ibm.com/analytics/cplex-optimizer> (2022年2月6日閲覧)
- [15] NTT データ数理システム, Numerical Optimizer, <https://www.msi.co.jp/nuopt/> (2022年2月6日閲覧)
- [16] M. Kuramata, R. Katsuki and K. Nakata, “Solving large break minimization problems in a mirrored double round-robin tournament using quantum annealing,” *PLoS ONE*, **17**, e0266846, 2022.
- [17] R. V. Rasmussen and M. A. Trick, “Round robin scheduling—a survey,” *European Journal of Operational Research*, **188**, pp. 617–636, 2008.
- [18] C. C. Ribeiro, “Sports scheduling: Problems and applications,” *International Transactions in Operational Research*, **19**, pp. 201–226, 2012.
- [19] J.-C. Régim, “Minimization of the number of breaks in sports scheduling problems using constraint programming,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **57**, pp. 115–130, 2001.
- [20] M. A. Trick, “A schedule-then-break approach to sports timetabling,” In *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling*, pp. 242–253, 2000.
- [21] H. L. Urdaneta, J. Yuan and A. S. Siqueira, “Alternative integer linear and quadratic programming formulations for ha-assignment problems,” In *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, **6**, 010311, 2018.
- [22] A. Suzuka, R. Miyashiro, A. Yoshise and T. Matsui, “The home-away assignment problems and break minimization/maximization problems in sports scheduling,” *Pacific Journal of Optimization*, **3**, pp. 113–133, 2007.
- [23] E. Farhi, J. Goldstone and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv*, arXiv:1411.4028, 2014.
- [24] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori and W. D. Oliver, “Perspectives of quantum annealing: methods and implementations,” *Reports on Progress in Physics*, **83**, 054401, 2020.

A. Appendix

2.2節で、式(5)の \mathcal{H}_1 における Z_i をスピン変数 $s_i \in \{-1, 1\}$ に置き換えると式(7)を導出できると述べた。この部分を説明する。

量子力学では、シュレディンガー方程式によって状態とエネルギーが定まる。特に、時間に依存しないシュレディンガー方程式は以下の固有方程式で表される。

$$\mathcal{H}|\psi\rangle = E|\psi\rangle \quad (24)$$

\mathcal{H} はハミルトニアンであり、エネルギーに関するエルミート演算子(エルミート行列)である。また、 E はエネルギーであり、実数値をとる。 $|\psi\rangle$ は量子状態(ベクトル)を表す。つまり、エネルギー E と量子状態 $|\psi\rangle$ は、ハミルトニアン \mathcal{H} の固有値と固有状態(固有ベクトル)となる。

1量子ビットの量子状態は、次の基底を用いて表される。

$$|+1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |-1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (25)$$

多くの文献では、 $|+1\rangle$ と $|-1\rangle$ はそれぞれ $|0\rangle$ と $|1\rangle$ で表されることが多いが、本稿では以下の説明の関係でこのような表記を用いる。このとき $s = \pm 1$ 対して、次の関係が成り立つ($\hat{\sigma}^z$ の定義は式(1))。

$$\hat{\sigma}^z |s\rangle = s |s\rangle. \quad (26)$$

式(24)のシュレディンガー方程式から、ハミルトニアン $\hat{\sigma}^z$ に対して、量子状態 $|s\rangle$ のエネルギーは s であることがわかる。 L 量子ビットの量子状態は、 $s_i = \pm 1$ ($i = 1, 2, \dots, L$)としたとき、次のように表すことができる。

$$|\psi\rangle = |s_1\rangle \otimes |s_2\rangle \otimes \dots \otimes |s_L\rangle \quad (27)$$

このとき、テンソル積の性質と式(26)より、次の関係が成り立つ(Z_i の定義は式(2))。

$$Z_i |\psi\rangle = s_i |\psi\rangle \quad (28)$$

これは、量子状態 $|\psi\rangle$ に対し、ハミルトニアン Z_i のエネルギーは s_i であることを示している。

以上の話を踏まえて、イジングモデルの量子力学的な表現のハミルトニアンと、エネルギーの関係を示す。式(28)より、次の関係が成り立つ。

$$\begin{aligned} \mathcal{H}_1 |\psi\rangle &= \left(-\sum_{i<j} J_{ij} Z_i Z_j - \sum_i h_i Z_i \right) |\psi\rangle \\ &= \left(-\sum_{i<j} J_{ij} s_i s_j - \sum_i h_i s_i \right) |\psi\rangle \end{aligned} \quad (29)$$

これより、スピン \mathbf{s} (量子状態は $|\psi\rangle = |s_1\rangle \otimes |s_2\rangle \otimes$

$\dots \otimes |s_L\rangle$) のエネルギーは $E(\mathbf{s}) = -\sum_{i<j} J_{ij} s_i s_j - \sum_i h_i s_i$ であることが確認できる. すなわち, ハミルトニアンを定義する Z_i をスピン変数 s_i に置き換えたものがエネルギーとなる.