Vol.2020-QS-1 No.20
2020/10/16

量子アニーリングシミュレータにおける疎行列表現方式の実行時間への影響

植田　圭[1,a)]　戸川　望[1]　木村　晋二[1]

概要：組み合わせ最適化問題の高速な解法として，近年量子アニーリングが注目を集めている．量子アニーリングでは，スピンとそれらの間の相互作用を考慮するイジングモデルを用い，最適化問題を系全体のエネルギーが最小となる問題に帰着する．この時，最大で量子数の二乗に比例した数の係数を入力として受け付ける必要があり，入力のデータサイズが大きなものとなる．係数行列は通常は疎行列であるので，非ゼロの係数のみを $(i,\ j,\ J_{ij})$ の形でインデックスをつけて表すが，それでも膨大なデータ量となるので，データ量を少なくする手法が必要になる．我々はこれまでに，量子アニーリングの入力となる相関係数の疎行列を，同じ値の繰返しと値の連続性に着目して，インデックス情報を削減できる手法をすでに報告している．具体的には，値の表にデータを二つずつ検索して繰り返し情報を含む一つのインデックス情報で登録する手法で，データの削減を行っている．本稿では，我々がこれまで提案したデータ構造をそのまま用いる量子アニーリングシミュレータの実行時間で評価結果を報告する．比較対象としては，通常の $(i,\ j,\ J_{ij})$ をそのまま記憶する手法を用いた．削減データ構造を用いる手法では，係数行列のサイズの大きい巡回セールスマン問題では実行時間が短くなることがわかった．70 地点の場合は 1.5 倍の高速化であった．また，その時のデータ量は約 1/25 であった．

キーワード：量子計算，イジングモデル，QUBO(Quadratic Unconstrained Binary Optimization)，巡回セールスマン問題，最小カット問題，データ圧縮

Effect of Sparse Matrix Representation Method on Execution Time in Quantum Annealing Simulator

Kei Ueda[1,a)]　Nozomu Togawa[1]　Shinji Kimura[1]

Abstract: Recently, quantum annealing has attracted attention as an efficient algorithm for combinatorial optimization problems. In quantum annealing, an optimization problem is expressed by using the ising model, which is a set of correlated quantums and their total energy is minimum corresponding to the optimal solution. The numeber of coefficients of the ising model is proportional to the square of the number of quantums, and the input data size becomes large. The coefficient matrix is usually a sparse matrix, so only the non-zero coefficients are indexed in the form $(i,\ j,\ J_{ij})$. Therefore, we are working on a method to reduce the size of sparse matrix representation. We have already reported a method that can reduce index information by reduced the repetition of the same value and the continuity of the values of the sparse matrix of the correlation coefficient in the input of the quantum annealing. As a results, the index information is represented the starting value and the number of repetitions. This time, we will check what the execution time of a quantum annealing simulator that uses this reduced data structure as it is. For the comparison, we used the method of storing ordinary $(i,\ j,\ J_{ij})$ as duplicated for $(i >= j)$ are. In the method using the reduced data structure, it was found that the execution time is an important factor in the traveling salesman problem with a large coefficient matrix size. At 70 points, the speed was 1.5 times faster. The amount of data at that time was about 1/25.

Keywords: quantum computer, ising model, QUBO (Quadratic Unconstrained Binary Optimization), traveling salesman problem, mincut problem, data compression

Vol.2020-QS-1 No.20
2020/10/16

1. はじめに

現実的な時間内で解くことが困難な組み合わせ最適化問題は多く存在する．これらの問題は計算機の性能向上により，解決してきた．しかし，計算機の性能向上にも限界が近づき，新たなアルゴリズムや計算機構が必要とされている [1][2]．そこで，現在注目されているのが，量子アニーリングに基づく計算機構である [3]．量子アニーリングでは複数の状態を持つ量子ビットを制御し，自らがエネルギーの低い状態に遷移する現象を利用することで，最適化問題を解く手法である．また，エネルギーが高くなる場合でも適当な確率で遷移させることで，局所最適解に落ち込むことを防ぎ，全域的な最適解を得ることができる．量子アニーリングは量子ビットを並列に扱うことで，類似手法のシミュレーテッドアニーリングに比べて，エネルギーの最小値を求める速度が指数関数的に速くなると言われている [4][5]．

量子アニーリングでは，解を求めたい最適化問題を磁性体の振る舞いを説明するための統計力学上のモデルであるイジングモデルにマッピングする．そして，スピンの数，相関係数の数，スピン間の相関係数 J_{ij} と自己エネルギー h_i で表す．これらの入力に対し，量子アニーリングコンピュータはシステム全体のエネルギーを最小化するようにスピンを決定する．また，量子モンテカルロ法を用いて，量子アニーリングコンピュータの動作の模擬を行う手法も知られている．量子モンテカルロ法では，量子のスピンのトグルをランダムに行い，システム全体のエネルギーを最小化する．量子モンテカルロ法は通常の計算機でも実行できるが，非常に多くの繰り返し処理に大きな計算時間を必要とするため，エミュレータを用いた高速化が有効である．しかし，エミュレータでは，通常使用できるメモリの量が限られているため，スピン数の二乗にもなる入力データの削減が必要になる．それに伴い，我々は Compressed Row Storage(CRS) 法 [6][7] をもとに，相関係数に 0 要素が多いことを活かした疎行列表現方式を報告している [10]．

本稿では，我々が提案している疎行列表現方式と，量子アニーリングシミュレータの実行時間との関連を評価した結果を報告する．今回は適用例として，巡回セールスマン問題と最小カット問題を扱った．巡回セールスマン問題は 70 都市，最小カット問題はノード数 1200 を最大として，C 言語を用いてデータ圧縮を行い，シミュレータで実行時間の評価を行う．

[1] 早稲田大学
Waseda University, 3-4-1, Okubo, Shinjuku, Tokyo, 169-8555 JAPAN
[†1] 現在，情報処理大学
Presently with Johoshori University
[a)] kei.ueda@islab.cs.waseda.ac.jp

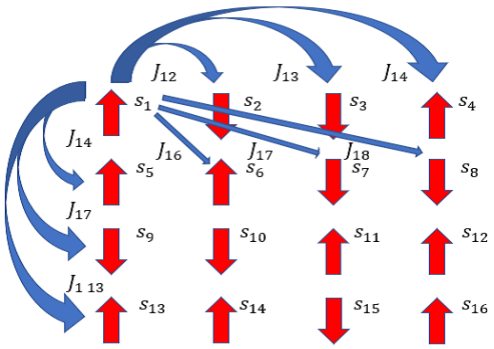

図 1　16 個のスピンと相互係数.

2. イジングモデルと QUBO

イジングモデルとは磁性を持つ量子の振る舞いを説明するための統計力学上のモデルである．s_i を i 番目の量子のスピンとする．スピンは ±1 のいずれかの値をとり，+1 のときは上向きスピン，-1 のときは下向きスピンである．さらに s_i，s_j 間の相互作用係数を J_{ij}，スピン s_i 一体にかかる外部磁場を h_i と表す [8][9]．J_{ij} は s_i，s_j が相互作用していないときは 0 となる．スピン数を n とするとイジングモデルのエネルギー関数 H は，

$$H = -\sum_{i<j} J_{ij} s_i s_j - \sum_{i}^{n} h_i s_i \qquad (1)$$

で定義される．実際の磁性体では，このエネルギー関数を最小化するようにスピンの向きが変化する．各種の最適化問題をこのイジングモデルで表すことで，実際の物理現象を用いて解くことができる．イジングモデルを実際に表したものを図 1 に示す．規則的に配置されたスピンにおいて左から 1 番目に配置されたスピンを s_1 と表し，2 番目に配置されたスピンを s_2 と順番に表していく．また，J_{ij} はスピン s_i と s_j の間の相互係数であり，スピン s_1 と s_2 間では J_{12} と表し，スピン s_1 と s_3 間においては J_{13} と表す．H の定義から，J_{ij} が正であると，s_i と s_j が同じ向きである場合にエネルギーが小さくなり．h_i が正である場合は s_i が上向きであるとエネルギーが小さくなる．

つぎに，QUBO とは Quadratic Unconstrained Binary Optimization の略で，1，0 を取る二値変数の値割当の最適化を考えるモデルである．QUBO は，変数の二次式で最適化のコストを表し，それが最小になるような変数への値割り当てを求める．また，QUBO をイジングモデルへ変換することも可能ある．QUBO の各変数 x_i をイジングモデルのスピン s_i に対応させ，以下の式に従って値を対応づけると，QUBO の最小解とイジングモデルのエネルギー最小の解が同じになる．

$$x_i = 0.5(s_i + 1) \qquad (2)$$

(2) 式から，イジングモデルのスピンが上向きで+1 をとるとき，QUBO の変数は+1 となる．同様に，イジングモデルのスピンが下向きで-1 をとるとき，QUBO の変数は 0 となる．イジングモデルの場合はスピンが 1 でも-1 でも加算しなければならないが，QUBO の場合は 1 の場合のみ加算すれば良いので演算回数を削減できる可能性がある．

3. 量子アニーリングのシミュレーション

ここでは，量子アニーリングを模擬するための量子モンテカルロ法 [1] について説明し，実行時間の比較対象となる二つのシミュレーション方式について述べる．

3.1 量子モンテカルロ法

量子アニーリングの模擬では，アドホックに決めた初期解から，系全体のエネルギーをコストとして，それが小さくなる方向に解の探索を行う．実際には量子モンテカルロ法を用いてランダムにスピンをトグルさせる．単純な近傍探索法のようにエネルギー H が小さくなる場合のみトグルさせる方式では，局所最適解に落ち込んでしまうからである．そこで量子アニーリングでは，量子揺らぎを用いて局所最適解に落ち込むことを防いでいる．量子揺らぎが高いときには現在の解よりも悪くなる解（すなわちエネルギーが大きい解）の方向の探索も積極的に受け入れる．一方量子揺らぎが低い時には，コストが下がる方向にしか探索しないようにすることで全域的な最適解が求まる．具体的には，エネルギーが減る場合は必ず変化させ，またエネルギーが大きくなる場合でも一定の確率で遷移させる．β を温度の逆数，ΔH をエネルギー変化前後の差分（変化後に大きくなるので正と仮定）として遷移確率は以下で表される．

$$\text{遷移確率} = \exp(-\beta \Delta H) \tag{3}$$

さらに，量子状態と呼ばれる複数の状態の重ね合わせを表すため，トロッタと呼ばれる複数のスピンの集合を扱う．異なるトロッタ間では，隣同士のトロッタ間でのみ温度に依存する相互係数に基づくエネルギー（横磁場に対応）を考える．系全体のエネルギーは以下のようになる [1].

$$\begin{aligned} H = \frac{1}{m}\sum_{k}^{m}(-\sum_{i}^{n}\sum_{j}^{n} J_{ij} s_{i,k} s_{j,k} - \sum_{i}^{n} h_i s_{i,k}) \\ -\frac{1}{2\beta}\log\coth(\frac{\beta\Gamma}{m})\sum_{k}^{m}\sum_{i}^{n} s_{i,k} s_{i,k+1} \end{aligned} \tag{4}$$

m はトロッタ数を表す．トロッタは，複数の状態から探索を行うことに対応している．k はトロッタの識別変数を表し，β は温度の逆数，Γ は横磁場の強さを表す．Γ を徐々に下げていくことで，$\frac{\beta\Gamma}{m}$ の項が小さくなって干渉が高まり，最小解に収束する．

量子モンテカルロ法による模擬は以下のように行う．

(1) トロッタ数分の初期スピンをランダムに決定する．
(2) Γ を初期化する．
(3) ランダムにトロッタとスピンを選び，それをトグルさせた場合のエネルギーの差分を計算する．
(4) エネルギーの差分に基づく遷移確率の式 (3) でスピンをトグルさせるかどうか決める．具体的には 0〜1 の間の乱数を発生させてそれが遷移確率より小さければトグルさせる．
(5) 規定の回数だけ (3)(5) を繰り返す．
(6) Γ を 0.99 倍するなどして小さくする．
(7) (3)(7) を規定回数繰り返す．

上記からわかるように，量子アニーリングの模擬は二重のループから構成される．内側のループをインナーループ，外側のループをアウターループという．インナーループは 10000 回以上，アウターループは 1000 回以上としている．量子アニーリングは最適解を保証できないが一般的に精度の高い解が得られることが知られている．

3.2 入力データについて

シミュレータの入力データについて述べる．まずは，巡回セールス問題における入力データをイジングモデル（実際は QUBO）で表した場合の変数とそれらの間の係数について述べる．巡回セールスマン問題では，都市を回る順序を表すために，都市と都市数分の時刻を掛けた変数空間を考える．時刻 t に都市 a を通る場合に $x_{t,a}$ が 1，そうでない場合に 0 とする．

都市 a と b 間の距離 $d^{a,b}$ が与えられているとすると，全経路長は L は，

$$L = \sum_{t,a,b} d^{a,b} x_{t,a} x_{t+1,b} \tag{5}$$

と定義される．また，全ての都市を 1 回ずつ訪問する条件を満たすために，2 つの制約条件を追加する．

(1) 同時刻にセールスマンはどこか 1 都市にしかいない．
(2) 全ての都市は一度ずつしか訪問しない．

まず前者の制約（1）に関しては，

$$(\sum_{a} x_{t,a} - 1)^2 \tag{6}$$

と表せ，これが全ての t で成り立たなければならないため，

$$\sum_{t}(\sum_{a} x_{t,a} - 1)^2 \tag{7}$$

となる．制約（2）に関しても同様に，

$$\sum_{a}(\sum_{t} x_{t,a} - 1)^2 \tag{8}$$

と表せる．したがって，目的関数全体 H は，

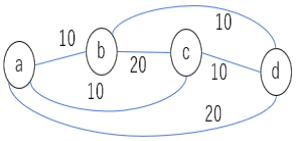

図 2 4 都市間の経路と距離.

```
16          1 6 20      2 7 10      4 8 200     7 8 20      10 11 200   -200
96          1 7 10      2 10 200    4 9 10      7 9 10      10 12 10    -200
0 1 200     1 9 200     2 12 10     4 10 10     7 10 10     10 13 20    -200
0 2 200     1 2 200     2 13 20     4 11 20     7 11 200    10 14 200   -200
0 3 200     1 3 200     2 14 200    4 12 200    7 15 200    10 15 10    -200
0 4 200     1 4 10      2 15 10     5 6 200     8 9 200     11 12 20    -200
0 5 10      1 5 200     3 4 20      5 7 200     8 10 200    11 13 10    -200
0 6 10      1 6 20      3 5 10      5 8 10      8 11 200    11 14 10    -200
0 7 20      1 7 10      3 6 10      5 9 200     8 12 200    11 15 200   -200
0 8 200     1 9 200     3 7 200     5 10 20     8 13 10     12 13 200   -200
0 12 200    1 12 10     3 11 200    5 11 10     8 14 10     12 14 200   -200
0 13 10     1 13 200    3 12 20     5 13 200    8 15 20     12 15 200   -200
0 14 10     1 14 20     3 13 10     6 7 200     9 10 200    13 14 200   -200
0 15 20     1 15 10     3 14 10     6 8 10      9 11 200    13 15 200   -200
1 2 200     2 3 200     3 15 200    6 9 20      9 12 10     14 15 200   -200
1 3 200     2 4 10      4 5 200     6 10 200    9 13 200                -200
1 4 10      2 5 20      4 6 200     6 11 10     9 14 20
1 5 200     2 6 200     4 7 200     6 14 200    9 15 10
```

図 3 4 都市の巡回セールスマン問題の入力データ.

$$H = \sum_{t,a,b} d^{a,b} x_{t,a} x_{t+1,b} + A \sum_{t} (\sum_{a} x_{t,a} - 1)^2 + A \sum_{a} (\sum_{t} x_{t,a} - 1)^2 \tag{9}$$

となる．A は正の定数であり，制約を満たすように適度に大きな値とする．この修正されたエネルギー関数 H をもとに係数を決める．まず (9) を展開し，$x_{t,a}^2 = x_{t,a}$ であることを用いて，自己エネルギー h_i は全てのスピンで $-2A$ となる．相互作用係数 J_{ij} は，自身とのコストを 0 とし，時刻 t+1 との距離 $d^{a,b}$ または $2A$ となる．

これらを踏まえ，図 2 に示す 4 都市の場合の QUBO の入力データの記述例を図 3 に示す．図 2 のグラフの枝のラベルが距離を表す．なお A としては 100 を用いる．入力として必要な情報は全部で 4 つある．1 つ目はスピンの数，2 つ目は 0 でない相互作用係数 J_{ij} の数である．3 つ目が相互作用係数 J_{ij}，4 つ目が自己エネルギー h_i である．この例では，4 都市であるので，スピン数は 16，J_{ij} の非ゼロ要素の数は 96，それから 96 個分の $(i,\ j,\ J_{ij})$ の並びとなる．また，このデータは削減前のデータなので,$i > j$ のみを記述している．その後，16 個分の h_i を並べて表す．J_{ij} と h_i は浮動小数点を用いるが，この例では整数値となっている．

次に，ノード数を均等に保つ最小カット問題についても述べる．最小カット問題とは，エッジに重みが付加されたグラフのノードを二つに分けた時に，異なるグループ間のエッジの重みの総和が最小になるように二分割する問題である．ノードの数が増えると計算量が膨大になり，NP 困難な問題に属する．ノード数 4，エッジ数 5 の場合のグラフの例を図 4 に示す．エッジに書かれた値が重みを表す．図ではいくつかの分割パターンを示している．分割するパターンは他にも存在するが，今回は三つの分割例を挙げた．左の場合のコストは，2+1+5=8．真ん中の場合は，2+1+3=6．右の場合は，5+1+7=13 となる．そのため，図 4 の場合だと真ん中を横に分割することで，最小値を得られ，最適解となる．一般的に，グラフ中のノードを二つのグループに分割する場合には，あるノードが所属するグループを表す二値変数 s_i を考え，最小化するエネルギー関数は,

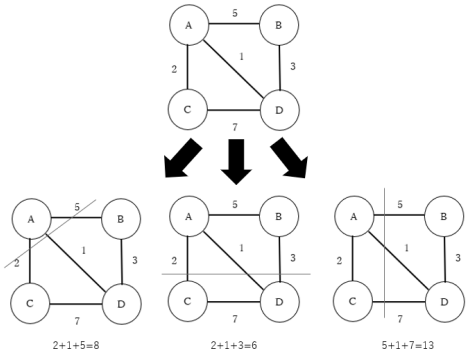

図 4 最小カット問題の例.

$$H = \sum_{(i,j)\in E} J_{ij}(s_i - s_j)^2 + \alpha(\sum_{i} s_i)^2 \tag{10}$$

となる．s_i が+1 をとるか，-1 をとるかで二つのグループ分けを行う．第一項では異なるグループのをつないでいるエッジのコストの総和を計算している．第二項はペナルティー項であり，グループ毎のノード数に偏りがあると罰則コストの α が働く．さらに (10) 式を QUBO に変換するため，(2) 式を代入する．

$$H = \sum_{(i,j)\in E} J_{ij}(x_i^2 + x_j^2 - 2x_i x_j) + \alpha[\sum_{i}\sum_{j} x_i x_j - N\sum_{i} + x_i^2] \tag{11}$$

なお，N はノード数である．

3.3 係数行列をそのまま用いたシミュレーション

量子アニーリングの入力は，最適化するエネルギーの計算に用いる係数 J_{ij} と h_i の値である．J_{ij} と J_{ji} は同じであるので，J_{ij} は $i < j$ の場合のみ示せばよい．

シミュレーションの実行上はスピン i をトグルさせたときにそれに関連するスピンとの J_{ij} をすべて求める必要があるので，$i > j$ の場合も重複してデータとして持つ方が効率が良い．そこで $i > j$ の場合のデータも重複して持つ方式のシミュレータを作成した．なお，重複させることでデータ量は 2 倍になる．各 i に対して重複させた $(i,\ j,\ J_{ij})$ を順番に処理することで，ΔH を求めることができる．CRS 法に基づき $(j,\ J_{ij})$ のデータを表す配列と各 i の先頭のデータを表す配列を表している．

```
16        031 00    741 01    822 00    1331 00   0     0 15 12
80        441 71    1123 51   1131 01   021 51    5     13
141 00    1131 61   021 71    1421 71   131 61    10    200.000000
521 10    1421 51   221 51    222 01    523 01    14    10.000000
721 21    021 01    431 00    521 21    931 61    18    20.000000
1221 01   221 11    821 71    721 51    1222 00   23    200.000000
1421 11   541 00    1021 51   922 00    1511 00   28    10.000000
022 00    921 10    1511 00   1241 11   031 11    32    10.000000
331 01    1121 21   024 00    331 61    622 01    38    200.000000
621 71    041 51    521 10    621 51    921 21    44    20.000000
923 01    422 00    722 21    831 00    1121 51   49    10.000000
1331 61   731 01    1031 00   1221 71   1322 00   54    10.000000
021 00    1021 71   1321 10   1421 51   021 71    59    200.000000
341 01    1311 00   1511 20   021 01    221 51    64    200.000000
723 51    041 11    123 01    231 11    731 61    70    -200.000000
1241 11   421 00    531 61    821 01    1021 51   75
                              1021 11   1231 00   81
                                                  1
```

図 5 4 都市の TSP における圧縮後の入力データ．

3.4 削減データを用いたシミュレーション

節 3.3 で表された入力データは J_{ij} は疎行列であることが多い．また，同じ値の繰り返しや，j の変化の規則性も見られる．そこで CRS 法に基づきつつインデックス情報を削減する手法を提案している [10]．これは i について，何番目のデータから i に関連するかの情報を別途持つことで，i の情報を除いて $(j,\ J_{ij})$ の 2 項組の並びで表すことのできる CRS 法に加え，値の表を導入して，入力データの値の連続性を用いて，j の情報も削減する手法である．例えば，図 3 からわかるように，(1，200)，(2，200)，(3，200)，(4，200) など同じ値が現れることがある．この場合，値表を別にして，そこへのインデックスで表すことで，値そのものを表すよりデータ量を削減できる．さらに j が順番に大きくなっているので，繰り返し回数を並記して，複数の $(j,\ J_{ij})$ を一度に表すことができる．また，値の表を別にすることで (5，10)，(6，10)，(7，20) に対する値の列は，10，10，20 のように j の値とは独立に入れられるので，$i = 4$ に対する (9，10)，(10，10)，(11，20) に対してもそれをそのまま使うことができる．今回は，値表の方でもインデックスを変更する必要があるので，(5，3，+1，値の先頭のインデックス，+1) と表す．5 が最初の j，3 が繰り返し回数，+1 が j の増分である．最後の+1 は値表側のインデックスを増やすかどうかのフラグである．先の例は，(1，4，+1，200 へのインデックス，+0) となる．しかし 4 都市の場合，図 3 と図 5 を比較してわかるようにデータ量は削減できていない．CRS と同様，提案手法では元のデータ量が小さい場合には，かえって記述量が増えるとに注意しなければならない．

この方式でのシミュレーションまでの手順を図 6 に示す．係数行列をそのまま用いる場合と同様，本削減法を用いても，$i > j$ の場合の重複を行った後に行う．$i < j$ の場合だけのデータと比較して，入力サイズが大きい場合はデータ量を大きく削減できることがわかっている．

4. 実行時間への影響

ここでは，3 章で述べた 2 つのシミュレーション方式における実行時間を比較する．また，メモリアクセス回数も測定することで，実行時間へどのように影響を及ぼしているか考察する．

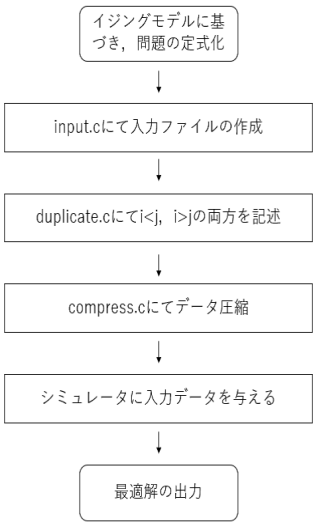

図 6 入力圧縮のフローチャート．

4.1 巡回セールスマン問題での評価

巡回セールスマン問題において 2 つのシミュレーション方式で実行時間を計測し，比較を行った．実験環境は，Intel Core i5-8400CPU@1.50GHz で 4.00GB 主記憶の計算機で Windows10 の Visual Studio で行った．各パラメータは β が 100.0，Γ が 1.0，トロッタ数が 30，インナーループが 10000，そしてアウターループを 1000 とした．実験方法は，都市数を 10 都市から最大で 70 都市まで変化させ，圧縮率の計算と実行時間を計測した．スピン数は 100〜4900 である．都市の $(x,\ y)$ 座標を整数でランダムに設定し，すべての都市間に座標に応じた最短の経路が存在するとして実験を行った．結果を表 1 に示す．表では削減前後のデータサイズとシミュレーションの実行時間を示している．なお，削減前のデータサイズは $i < j$ の場合のもので，削減後は $i < j$，$i > j$ の両方を含む場合である．また，(1，1)，(2，2)，(3，3)，... のように規則的に設定した場合の結果を表 2 に示す．圧縮率に関しては，都市数を増やすごとに高くなり，70 都市の場合で，4.60% まで削減された．また実行時間に関しても，圧縮率が高い方が実行時間が高速化され，特に規則的に設定した 70 都市の場合で，1.83 倍の高速化が実現された．

4.2 最小カット問題での評価

最小カット問題でも巡回セールスマン問題と同じ実験環境で評価を行った．巡回セールスマン問題では都市数 n に対して，スピン数は n^2 になるが，最小カット問題はノード数 n がスピン数となる．ノード間のエッジの重みをランダムに整数値を設定し，ノード数を 10 から 1100 の間で圧縮

表 1 巡回セールスマン問題での実行時間の比較 (ランダム).

都市数	ランダム		削減率 [%]	実行時間 [s]		高速化 [倍]
	削減前 [Byte]	削減後 [Byte]		duplicate	compress	
10	32,333	11,371	35.17	5.23	4.14	1.26
20	306,014	50,848	16.62	7.05	6.41	1.10
30	1,057,834	119,562	11.30	7.78	6.64	1.17
40	2,642,856	216,919	8.21	10.90	9.20	1.18
50	5,312,156	344,224	6.48	12.57	9.51	1.32
60	9,312,616	499,455	5.36	15.32	10.60	1.45
70	14,918,236	686,223	4.60	18.83	12.24	1.54

表 2 巡回セールスマン問題での実行時間の比較 (規則的).

都市数	規則的		削減率 [%]	実行時間 [s]		高速化 [倍]
	削減前 [Byte]	削減後 [Byte]		duplicate	compress	
10	32,313	11,303	34.98	7.83	5.04	1.55
20	303,574	50,541	16.65	9.83	6.82	1.44
30	1,057,234	116,948	11.06	12.59	8.75	1.44
40	2,633,896	214,895	8.16	13.98	11.61	1.20
50	5,292,216	344,064	6.50	15.98	12.69	1.26
60	9,292,216	499,592	5.38	18.92	12.72	1.49
70	14,905,344	685,022	4.60	27.63	15.06	1.83

率と実行時間を評価した．結果をまとめたものを表 3 に示す．最小カット問題では，巡回セールスマン問題に比べ，圧縮率は最大で約 90% とあまり良い結果が得られなかった．原因は，最小カット問題の J_{ij} に対応するエッジの重みをランダムに決めているため，同じ値の出現頻度が少ないことが考えられる．ランダムな重みで問題を作成する場合は今の圧縮率が限界と考える．しかし，圧縮率はあまり良くないものの，実行時間に関しては，ノード数 200 以上の場合でも常に高速化が実現された．特に，ノード数 200 の時に最大で 1.77 倍の高速化された．

4.3 メモリアクセス回数との関連

ここでは，削減データを用いたシミュレーション手法と実行時間の関係について，アニーリング実行時のメモリアクセス回数をもとに考察する．アクセス数をカウントした配列は係数値の配列 val_ary，スピンの配列 s，i 番目のスピンに対し $(j,\ J_{ij})$ の先頭のインデックスを表す top_J の 3 つである．val_ary 値表の数である．例として，図 5 の最後の列にある相関係数の連続した値に当たる．s はスピンの初期解を格納する配列である．スピン i をトグルする場合は，まず top_J をアクセスし，$(j,\ J_{ij})$ の情報を参照する．削減法の説明で述べたように，$(j,\ J_{ij})$ は繰り返しと val_ary へのインデックスで表している．その情報に基づき，val_ary にアクセスする．最後に，top_J はスピン i の先頭の値が何番目にあるかの情報格納した配列である．この top_J の値は圧縮率が高いとそのアクセス数を削減することができる．これを踏まえ，巡回セールスマン問題のランダムに座標を設定した場合の結果を表 4 に，規則的に座標を設定した場合を表 5 に示す．また，最小カット問題のメモリアクセス回数を表 6 に示す．削減率は係数行列をそのまま用いたシミュレーションの合計のアクセス回数と削減データを用いたシミュレーションの合計アクセス回数から算出している．

巡回セールスマン問題は，最小カット問題に比べてデータの削減率が高く，それにより top_J へのアクセス回数が大きく削減でき，どの都市数でも高速化が実現した．最小カット問題に関しては，アクセス回数は削減されたものの，すべてが高速化という結果は得られていない．これらの結果から，データ削減により top_J のアクセス回数を削減でき，高速化を実現することができる．しかし，データの削減率が低い場合でもアニーリング実行時の各配列のアクセス回数が削減されていれば，高速化の可能性があるということがわかった．

5. おわりに

本稿では，データ削減による実行時間への影響について述べた．データの削減率が高いと，アニーリング実行時にメモリアクセス回数が減り，高速化が実現できる．しかし，最小カット問題のようにデータ削減率が悪い問題でもメモリアクセス回数は削減されていれば，高速化が実現される場合がある．

今後はこの結果を踏まえ，まずは，圧縮法の改良を検討する必要がある．特に，最小カット問題のような値の連続性があまりない問題に対しても削減率を高める手法を考える．また，アニーリング実行時の各配列のアクセス回数についてより詳細に調べることで，実行時間との関連性をよ

表 3 最小カット問題での実行時間の比較.

ノード数	ランダム		削減率 [%]	実行時間 [s]		高速化 [倍]
	削減前 [Byte]	削減後 [Byte]		duplicate	compress	
10	845	1,604	189.82	5.24	3.90	1.34
20	3,463	5,486	158.42	6.03	5.25	1.15
30	7,852	11,592	147.63	6.68	6.59	1.01
40	13,999	19,649	140.36	7.05	7.41	0.95
50	21,933	30,410	138.65	7.61	8.45	0.90
60	31,623	43,829	138.60	8.31	9.59	0.87
70	43,124	59,046	136.92	9.43	10.47	0.90
80	56,378	76,420	135.55	10.39	11.08	0.94
90	71,406	96,775	135.53	10.73	12.52	0.86
100	88,223	119,356	135.29	11.54	13.16	0.88
200	373,832	463,143	123.89	21.30	12.06	1.77
300	856,797	996,527	116.31	31.23	17.58	1.78
400	1,537,097	1,701,555	110.70	41.13	26.02	1.58
500	2,414,595	2,573,090	106.56	52.33	35.06	1.49
600	3,489,441	3,584,878	102.74	63.16	49.98	1.26
700	4,761,674	4,690,608	98.51	74.10	64.51	1.15
800	6,231,310	5,977,279	95.92	84.71	75.76	1.12
900	7,898,051	7,394,403	93.62	97.84	85.09	1.15
1000	9,762,428	8,948,327	91.66	109.23	96.30	1.13
1100	11,933,982	10,700,290	89.66	117.58	116.73	1.01

表 4 巡回セールスマン問題でのアクセス回数 (ランダム).

都市数	duplicate				compress				削減率 [%]
	val_ary	s	topJ	合計	val_ary	s	topJ	合計	
10	35,984,527	55,962,529	359,604,036	451,551,092	45,976,229	46,975,801	52,842,814	145,794,844	32.29
20	38,265,729	58,243,731	759,164,076	855,673,536	48,251,065	48,752,925	56,462,379	153,466,369	17.94
30	39,802,615	59,780,617	1,158,724,116	1,258,307,348	49,828,598	50,171,902	58,899,924	158,900,424	12.63
40	41,985,956	61,963,958	1,558,284,156	1,662,234,070	51,980,158	52,248,848	58,681,427	162,910,433	9.80
50	45,272.484	65,250,486	1,957,844,196	2,068,367,166	55,282,711	55,513,546	58,530,723	169,326,980	8.19
60	50,247,235	70,225,237	2,357,404,236	2,477,876,708	60,462,427	60,674,737	58,776,468	179,913,632	7.26
70	57,195,960	77,173,962	2,756,964,276	2,891,334,198	67,236,160	67,443,353	58,939,987	193,619,500	6.70

り明確にし，さらなる高速化を実現したい.

謝辞 早稲田大学・情報システム研究室の柳澤政生教授，史又華教授，吉増敏彦教授はじめメンバーの皆様には日頃からのご討議を心から感謝します．また，早稲田大学戸川研究室の田中宗准教授他メンバーの皆様にも日頃からのご討議に感謝します．この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）委託業務の結果，得られたものである．また，NEDO のプロジェクトを総括いただく、NEC のご担当の皆様に感謝します.

参考文献

[1] Takuya Okuyama, Masato Hayashi, and Masanao Yamaoka “An Ising computer based on simulated quantum annealing by path integral Monte Carlo method,” *Proc. of IEEE International Conference on Rebooting Computing*, pp.1-6, November 2017.

[2] Jasmeet Singh and Mohit Singh “Evolution in Quantum Computing,” *Proc of IEEE International Conference System Modeling & Advancement in Research Trends*, pp.1-4, November 2016.

[3] 塚本 三六, 高津 求, 松原 聡, 田村 泰孝“組み合わせ最適化問題向けハードウェアの高速化アーキテクチャー,” *Fujitsu Vol.68*, pp.8-14, June 2017.

[4] Elizabeth Crosson, Aram W. Harrow “Simulated Quantum Annealing Can Be Exponentially Faster than Classcal Simulated Annealing,” *Proc. of IEEE 57th Annual Symposium on Foundations of Computer Science*, pp.1-10, October 2016.

[5] Pierre Berg，Baptiste Cavarec，Arpad Rimmel，Joanna Tomasik “Restricting the search space to boost Quantum Annealing perfoemance,” *Proc. of IEEE Congress on Evolutionary Computation*, July 2016.

[6] 曽我 尚人，佐藤 真平，中原 啓貴“Sparse Robust Autoencoder による心電図外れ値検出器のハードウェア向けのモデル圧縮について，” 信学技報 *IEICE Technical Report VLD2018-114*，February 2019.

[7] Ryan Kastner, Janarbek Matai, and Stephen Neuendorffer “Parallel programming for fpgas.” *CoRR*, Vol. abs/1805.03648, 2018.

[8] 金丸 翔, 於久 太祐, 多和田 雅師, 田中 宗, 林 真 人, 山岡 雅直, 柳澤 政生, 戸川 望, “イジング計算機によるスロット配置問題の解法, ” 信学技法 *VLD2018-34*, Vol. 118,

表 5 巡回セールスマン問題でのアクセス回数 (規則的).

都市数	duplicate				compress				削減率 [%]
	val_ary	s	topJ	合計	val_ary	s	topJ	合計	
10	35,984,544	55,962,546	359,604,036	451,551,126	45,975,829	46,975,681	52,842,814	145,794,324	32.29
20	38,265,549	58,243,551	759,164,076	855,673,176	48,251,416	48,755,159	56,412,828	153,419,403	17.93
30	39,803,279	59,781,281	1,158,724,116	1,258,308,676	49,828,865	50,172,151	57,593,008	157,594,024	12.52
40	41,985,749	61,963,751	1,558,284,156	1,662,233,656	51,979,148	52,247,728	58,180,198	162,407,074	9.77
50	45,273.210	65,251,212	1,957,844,196	2,068,368,618	55,283,312	55,514,223	58,530,723	169,328,258	8.19
60	50,248,220	70,226,222	2,357,404,236	2,477,878,678	60,463,153	60,675,802	58,765,234	179,904,189	7.26
70	57,194,810	77,172,812	2,756,964,276	2,891,331,898	67,236,638	67,443,951	58,933,926	193,614,515	6.70

表 6 最小カット問題でのアクセス回数.

ノード数	duplicate				compress				削減率 [%]
	val_ary	s	topJ	合計	val_ary	s	topJ	合計	
10	44,950,299	64,928,301	89,901,009	199,779,609	54,952,884	59,933,970	26,964,514	141,851,368	71.00
20	94,577,970	114,555,972	189,791,019	398,924,961	104,906,976	109,869,030	28,967,671	243,743,677	61.10
30	144,520,310	164,498,312	289,681,029	598,699,651	155,154,560	160,080,410	40,619,292	355,854,262	59.44
40	193,813,360	213,791,362	389,571,039	797,175,761	205,046,362	210,038,594	48,699,405	463,784,361	58.18.
50	245,420,464	265,398,466	489,461,049	1,000,279,979	253,741,243	258,715,905	57,137,650	569,594,798	56.94
60	295,327,625	315,305,627	589,351,059	1,199,984,311	303,554,182	308,552,783	54,262,466	666,369,431	55.53
70	344,623,540	364,601,542	689,241,069	1,398,466,151	354,593,096	359,602,227	59,649,531	773,844,854	55.34
80	394,603,906	414,581,908	789,131,079	1,598,316,893	404,248,646	409,216,969	65,796,851	879,262,466	55.01
90	444,837,839	464,815,841	889,021,089	1,798,674,769	454,753,005	459,751,932	61,813,859	976,318,796	54.28
100	495,110,867	515,088,869	988,911,099	1,999,110,835	504,193,154	509,178,398	57,623,591	1,070,995,143	53.57
200	994,894,722	1,014,872,724	1,987,811,199	3,997,578,645	705,500,774	711,293,292	134,978,774	1,551,772,840	38.82
300	1,492,702,061	1,512,680,063	2,986,711,299	5,992,093,423	1,064,096,533	1,072,484,391	316,037,710	2,452,618,634	40.93
400	1,992,869,951	2,012,847,953	3,985,611,399	7,991,329,303	1,579,172,803	1,586,739,160	581,414,92,	3,747,326,891	46.89
500	2,491,594,947	2,511,572,949	4,984,511,499	9,987,679,395	2,001,046,858	2,008,019,143	909,528,627	4,918,594,628	49.25
600	2,992,031,761	3,012,009,763	5,983,411,599	11,987,453,123	2,490,006,779	2,496,448,058	1,315,866,265	6,302,321,102	52.57
700	3,491,480,279	3,511,458,281	6,982,311,699	13,985,250,259	2,939,129,850	2,945,246,704	1,835,762,768	7,720,139,322	55.20
800	3,990,629,011	4,010,607,013	7,981,211,799	15,982,447,823	3,339,487,257	3,345,291,356	2,306,920,411	8,991,699,024	56.26
900	4,490,014,582	4,509,992,584	8,980,111,899	17,980,119,065	3,735,195,370	3,740,821,059	2,793,475,982	10,269,492,411	57.12
1000	4,988,780,083	5,008,758,085	9,979,011,999	19,976,550,167	3,964,910,428	3,970,052,989	3,306,202,631	11,241,166,048	56.27
1100	5,489,646,318	5,509,624,320	10,977,912,099	21,977,182,737	4,549,780,486	4,554,860,049	3,812,680,164	12,917,320,699	58.78

No. 83, pp.161-166, June 2018.

[9] Masanao Yamaoka, Chihiro Yoshimura, Masato Hayashi, Takuya Okuyama, Hidetaka Aoki, and Hiroyuki Mizuno “A 20k-Spin Ising Chip to Solve Combinatorial Optimization Problems With CMOS Annealing,” *Proc. of IEEE JOURNAL OF SOLID-STATE CIRCUITS*, VOL. 51, NO. 1, January 2016.

[10] 植田 圭，戸川 望，木村 晋二，”量子アニーリングエミュレータのためのデータ構造，”情報処理学会，DA シンポジウム 2019 論文集，pp.39-44，August 2019.