

複雑な制約を持つ最適化問題における
量子アニーリングの効果的な利用方法の研究

山梨大学大学院 医工農学総合教育部 博士課程学位論文
2024年 3月
小見山 朋子

目次

第 1 章	序論	1
1.1	はじめに	1
1.2	研究成果	3
1.3	本論文の構成	4
第 2 章	量子アニーリングと現状の課題	5
2.1	量子アニーリング	5
2.2	量子アニーリングにおける現状の課題	7
第 3 章	疎行列の直接解法向けフィルイン削減オーダリング	10
3.1	疎行列のオーダリング	10
3.2	先行研究	11
第 4 章	Quantum minimum fill-in algorithm	15
4.1	QUBO 定式化	15
4.2	高次式の次数削減	18
4.3	不等式の利用	20
4.4	パラメータチューニング	28
4.5	評価	33
第 5 章	QMF の問題緩和とハイブリッドオーダリング	42
5.1	問題緩和	42
5.2	QMF と MD のハイブリッドオーダリング	49
第 6 章	結論	52
6.1	まとめ	52
6.2	今後の課題	52
	参考文献	55
	関連文献	60

付録 A	Domain-wall encoding による QMF の QUBO 定式化	61
A.1	QUBO 定式化	62
A.2	評価	65
A.3	QUBO 行列の解析	66
付録 B	不等式の QUBO 変換手法の比較	72
B.1	ナップザック問題	72
B.2	集合被覆問題	73
B.3	時間制約付き経路最適化	76

目次

1.1	ムーアの法則 [1]	2
2.1	4 変数イジングモデルの例	6
2.2	3 変数イジングモデルの例	6
2.3	Chaininig[22]	9
3.1	消去グラフの例	11
3.2	MD の適用例 [a]	12
3.3	ND の適用例	13
3.4	並列処理向け CM・RCM の適用例	13
4.1	QMF の変数設定例	15
4.2	制約 1 の変数値例	16
4.3	制約 2 の辺と頂点	17
4.4	式 (4.3) を利用した場合のハミルトニアン全体のエネルギー値	18
4.5	QMF の QUBO の変数の総数 [a]	26
4.6	QMF の実行可能解数 [a]	26
4.7	MaxSAT の QUBO の変数の総数 [a]	27
4.8	QMF の QUBO の変数の総数 [a]	28
4.9	QMF の実行可能解数 [a]	28
4.10	MaxSAT の QUBO の変数の総数 [a]	29
4.11	5 × 5 行列に対する消去順を固定した消去グラフの例	31
4.12	5 × 5 行列に対する最適解率	32
4.13	6 × 6 行列に対する最適解率	32
4.14	7 × 7 行列に対する最適解率	33
4.15	5 × 5 行列に対する MD のフィルイン数以下の解率	34
4.16	6 × 6 行列に対する MD のフィルイン数以下の解率	34
4.17	7 × 7 行列に対する MD のフィルイン数以下の解率	35
4.18	5 × 5 行列に対する局所解率	35
4.19	6 × 6 行列に対する局所解率	36

4.20	7 × 7 行列に対する局所解率	36
4.21	フィルイン数の比較	40
4.22	計算時間の比較	41
5.1	最初と最後の半分のステップでそれぞれ発生する平均フィルイン数 [b]	43
5.2	ハイブリッドオーダリングの QMF における変数設定例 [b]	44
5.3	計算に必要な変数の総数の平均数 [b]	45
5.4	実行可能解の総数 [b]	46
5.5	次数削減を利用した QUBO(5.2) を用いた QMF と他手法の平均フィルイン数 [b]	47
5.6	不等式を利用した QUBO(5.3) を用いた QMF と他手法の平均フィルイン数 [b]	48
5.7	総当たりアルゴリズムの計算時間	50
5.8	ハイブリッドオーダリングにおける QMF のオーダリング対象頂点	50
5.9	各オーダリング手法における平均フィルイン数	51
5.10	各オーダリング手法における平均計算時間	51
A.1	One-hot encoding と Domain-wall encoding	61
A.2	QMF の変数設定例 (Domain-wall encoding)	62
A.3	制約 1 の変数例 (Domain-wall encoding)	63
A.4	One-hot encoding と Domain-wall encoding による QMF の変数の数	65
A.5	One-hot encoding と Domain-wall encoding による QMF の変数の平均接続数	66
A.6	One-hot encoding と Domain-wall encoding の実行可能解が得られた確率の比較	67
A.7	One-hot encoding と Domain-wall encoding の最適解が得られた確率の比較	67
A.8	各 QUBO 行列の最小固有値	68
A.9	各 QUBO 行列の最大固有値	68
A.10	各 QUBO 行列の固有値範囲	69
A.11	各 QUBO 行列の最小特異値	70
A.12	各 QUBO 行列の最大特異値	70
A.13	各 QUBO 行列の特異値範囲	71
A.14	各 QUBO 行列の条件数	71
B.1	ナップザック問題に対する計算に必要な変数の総数	73
B.2	ナップザック問題の実行可能解が得られた確率	74
B.3	ナップザック問題の価値の総和	74
B.4	2 次ナップザック問題の実行可能解が得られた確率	75

B.5	2次ナップザック問題の価値の総和	75
B.6	集合被覆問題に対する計算に必要な変数の総数	76
B.7	集合被覆問題の実行可能解が得られた確率	76
B.8	集合被覆問題の解のコスト	77
B.9	時間制約付き経路最適化の実行可能解が得られた確率	78

表目次

2.1	現状のアニーリングマシン [14]	8
4.1	次数削減後の変数の総数	21
4.2	パラメータの範囲と計算に必要な量子ビット数	30
4.3	次数削減による QMF の結果	37
4.4	不等式制約による QUBO の結果	38
4.5	変数の接続数	39
5.1	次数削減を利用した QUBO(5.2) を用いた QMF と他手法のフィルイン数の比較 [b]	47
5.2	不等式を利用した QUBO(5.3) を用いた QMF と他手法のフィルイン数の比較 [b]	48
A.1	One-hot encoding と Domain-wall encoding の比較 [55]	62

第 1 章

序論

1.1 はじめに

コンピュータの性能は、半導体の集積率が 18 ヶ月で 2 倍になるというという「ムーアの法則」(図 1.1) に従って向上してきた。しかし、近年集積回路の回路線幅が原子レベルに近づいてきており、いま以上に高集積度の回路の作成は難しい、つまり、性能向上の限界が近づいている。さらに、スーパーコンピュータをはじめとする大規模演算装置は膨大な電力を消費し、今後さらに発展するであろう IT 社会のエネルギー不足が大きな課題になると予想される。そこで、近年、量子の性質を利用して計算を行う量子コンピューティング技術が注目を集めている。この技術は、一部の計算においては従来のコンピュータよりも高速に計算ができる可能性がある。従来のコンピュータが“0”と“1”の(古典)ビットで計算を行うのに対し、量子コンピュータは“0”と“1”に加えて“0と1の重ね合わせ状態”をもつ量子ビットを利用して計算を行う。“0と1の重ね合わせ状態”とは、量子力学の法則に基づくものであり、外部から観測されて“0”か“1”かが決定されるまでは、“0”でもあり“1”でもある状態を意味する。この重ね合わせ状態を利用することで、量子コンピュータは NP 困難な問題などを古典コンピュータよりも高速に解くことができる可能性がある。さらに、量子コンピュータは、量子ビットに超電導体を利用することで、CPU とメモリの機能を持つ 1 平方センチ程度のチップを極低温に冷やすため以外にはほとんど電力が必要にならない [2]。例えば、量子コンピュータの 1 つである D-Wave 社の量子アニーリングマシンの消費電力量は、スーパーコンピュータ富岳の消費電力の 1500 分の 1 程度である [2, 3]。これらの利点から、今後の IT 社会を支える技術として量子コンピューティング技術が重要視されている。

量子コンピューティング技術の開発は世界中で進められている。現在、応用研究まで活発に進められている主な技術は、「量子ゲート方式」と「量子アニーリング方式」の 2 つである。量子ゲート方式は、量子論理ゲートを利用して計算を行う量子コンピュータである。汎用的な利用が可能であるが、ノイズの影響を受けやすく、現在のところ極少ビット数までしか実装されていない。日本国内では、2023 年 3 月に 64 量子ビットの国産量子コンピュータの初号機が稼働を開始 [4] しており、これに続く国産量子コンピュータの開発も始まっている。一方、1998 年に西森ら [5] によって提唱された日本生まれの技術である量子アニー

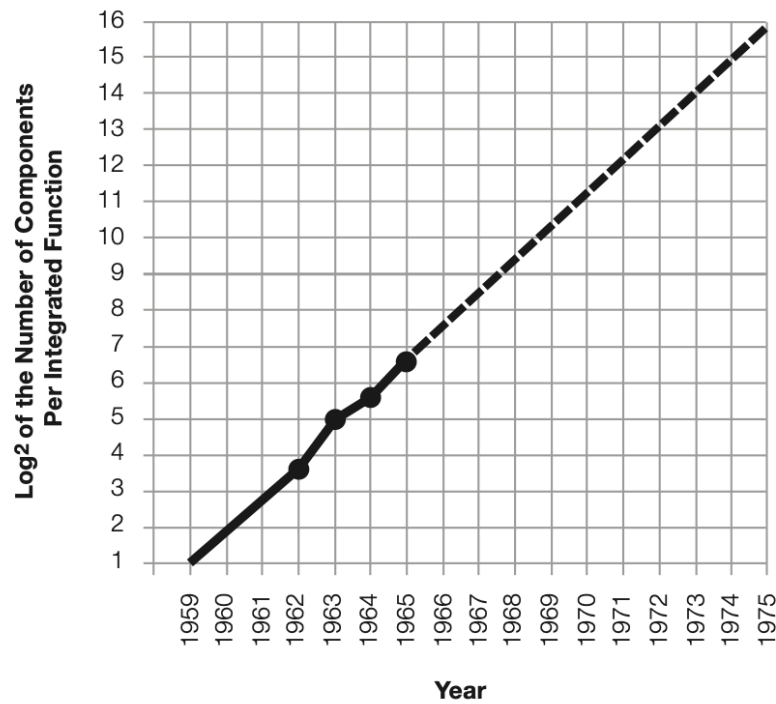


図 1.1 ムーアの法則 [1]

リング方式は、エネルギー曲線上を量子の性質を利用して探索を行い、組合せ最適化問題の最適解を得ることができる。ノイズに対してある程度の耐性があり、現在、シミュレータも含め実用的な規模の問題に適用され始めている。また、量子アニーリング方式で計算できることは、量子ゲート方式でも同等の効率で計算ができることが理論的に証明されている [2]。特に、量子ゲート方式の分野で近年活発に研究されている量子近似最適化アルゴリズム (Quantum Approximate Optimazation Algorithm; QAOA) は、量子アニーリング方式と同様に組合せ最適化問題を解くことが可能である。量子コンピューティング技術が未だ開発段階にあることもあり、これらの技術の応用研究は始まったばかりである。そのため、これらの技術をどのような用途で利用すればよいのか、また、課題がどこにあるのかがあまり明確になっていない。本研究では、量子アニーリングの応用利用における課題と対処法を議論する。

科学技術計算において、疎行列連立一次方程式の求解は頻繁に登場する。疎行列とは、行列要素のほとんどが零である行列のことであり、非零要素のみを扱うことでメモリの消費量や計算コストを削減することができる。このような行列を含む方程式を直接解法で解くと、零だった要素が非零になるフィルインが発生し、計算コストなどが増加する問題がある。この問題に対して、あらかじめ行列の要素を並べ替えることでフィルインの数を減らす前処理手法であるオーダリングが利用されてきた。このアルゴリズムとして様々なヒューリスティクスが研究されてきたが、2021年に小見山 [6] は、新しい解法として、量子アニーリング (Quantum annealing; QA) を利用してオーダリングを得る Quantum minimum fill-in (QMF) algorithm を提案した。量子コンピューティング技術が、数値計算分野に応用され

た例はここ 1~2 年で数件の発表があった程度であり、2021 年時点では、[6] の研究は数値計算分野に QA を適用したおそらく世界で初めての例であった。この研究では、QMF の提案と定式化、日立 CMOS アニーリングマシン [7] による評価を行ったが、新たな QA マシンの開発や応用研究の発展がここ数年で大きく進んだこともあり、現在の知識からは定式化方法と評価が不十分なものであった。本論文では、疎行列のオーダリングへの QA の適用を例に、QA に適した複雑な制約を持つ問題に対する定式化手法と QA の効果的な利用方法の提案を目的に実施された研究結果をまとめる。

1.2 研究成果

本研究の 4 つの成果をそれぞれ説明する。

- **QA を利用した疎行列のオーダリング手法である QMF の QUBO 定式化と評価 [a]**

最小フィルインを求める疎行列のオーダリング手法である QMF を二次制約なし二値最適化 (Quadratic Unconstrained Binary Optimization; QUBO) の形に定式化し、QA で従来手法よりも少ないフィルイン数の解が得られる可能性を示した。従来、疎行列のオーダリングは動的な発見的手法で求められてきたが、本研究では、疎行列のオーダリングを静的な組合せ最適化問題として定式化し、QA などの最適化技術により解を求めることを可能にした。

- **QA に適した複雑な制約を持つ問題の QUBO 定式化 [a, d]**

複雑な制約を持つ問題を少ない変数で QUBO 定式化するための手法を、高次式の次数削減と不等式制約の利用の 2 つの観点から議論、評価した。特に、不等式を利用した定式化は新たに提案した手法であり、変数が少ないだけでなく、従来利用されてきた高次式の次数削減よりも実行可能解が得られやすい可能性があることを示した。

- **QA におけるパラメータチューニングの重要性を議論 [c]**

QA で問題を解く際には、「QUBO に定式化した際のコスト関数と制約関数の重み」と「QA マシンのハードウェアパラメータ」の 2 種類のパラメータチューニングが必要になる。どちらも QA で質の良い解を得るために重要であるが、本研究では、前者のパラメータチューニングの重要性を示した。

- **QA マシンを効果的に利用する手法の提案 [b]**

現状の QA マシンを効果的に利用する手法として、QMF の問題緩和と QA と古典アルゴリズムのハイブリッド手法を議論した。現状の QA マシンには、実用上の課題が多くあり、大規模な問題を QA マシン単体で解くことはあまり現実的ではない。そこで、QA の利点を生かしつつ、QA マシンの課題を古典コンピュータによりフォローすることで、従来手法よりもよい結果を得ることを示した。

1.3 本論文の構成

本論文の残りの部分の構成は次の通りである。2章で、本研究で扱う量子コンピューティング技術である量子アニーリングとその利用上の課題を説明し、3章で、疎行列の直接解法向けフィルイン削減オーダリングの概要と従来手法を説明する。4章で、量子アニーリングを利用して疎行列のオーダリングを求める手法である Quantum minimum fill-in (QMF) algorithm の導入と、少ない変数で定式化する手法の提案と評価を行う。さらに、QMF の評価結果を示す。5章で、QMF の問題緩和手法と QMF と従来のオーダリング手法である Minimum degree algorithm のハイブリッドオーダリングの提案と評価を行う。7章で、本研究のまとめと今後の展望を述べる。

第 2 章

量子アニーリングと現状の課題

2.1 量子アニーリング

量子アニーリング (Quantum Annealing; QA) は, 1998 年に西森ら [5] によって提唱された技術である. 量子ゆらぎ効果による状態間の遷移を利用して解の探索を行う, 組合せ最適化問題に対する新たなメタヒューリスティック解法として提唱された. 以下参考文献 [2] を基に QA について説明する.

イジングモデルは, 格子点上にスピンを配置した構造を持つ磁性体のもっとも簡単なモデルである. 量子アニーリングの原理を理解するうえで起点となるのは次式のイジングモデルのハミルトニアンである.

$$H(\sigma) = - \sum_{i < j} J_{ij} \sigma_i \sigma_j - \sum_{i=1}^N h_i \sigma_i \quad (2.1)$$

ここで, $\sigma_i \in \{-1, 1\}$ はスピン変数, J_{ij} は σ_i と σ_j の間の相互作用, h_i は σ_i の局所磁場である. 相互作用 J_{ij} は, 正のとき ($J_{ij} > 0$) は $\sigma_i = \sigma_j$, 負のとき ($J_{ij} < 0$) は $\sigma_i = -\sigma_j$ となるように σ_i, σ_j に力を働かせるものである. 一方, 局所磁場は, h_i と σ_i の符号が逆になるように σ_i に力を働かせる. また, ハミルトニアンの値 (エネルギー値) が最小となるスピンの状態を基底状態と呼ぶ. 二次平面上の正方格子点上にイジング変数が配置された場合の相互作用と局所磁場の例を図 2.1 に示す. この図では, イジング変数 σ_i と σ_j の間に相互作用 $J_{ij} > 0$, σ_k と σ_l の間に相互作用 $J_{kl} < 0$ が働いている. さらに, イジング変数 σ_i と σ_l にそれぞれ局所磁場 $h_i > 0$ と $h_l < 0$ が働いている. このイジングモデルの基底状態は, $\{\sigma_i, \sigma_j, \sigma_k, \sigma_l\} = \{-1, -1, -1, 1\}$ である.

図 2.1 は基底状態が自明であるが, 図 2.2 に示すように全ての相互作用エネルギー (式 (2.1) の第 1 項目) を最小化することができないような場合もある. 図 2.2 では, 3 つのイジング変数がそれぞれ異なる値になるように相互作用がかかっている. しかし, そのような解は存在せず, ハミルトニアン全体の基底状態として, いずれか 1 つの相互作用エネルギーが基底状態ではない 6 通りの状態が得られる. このように, 局所的な基底状態とハミルトニ

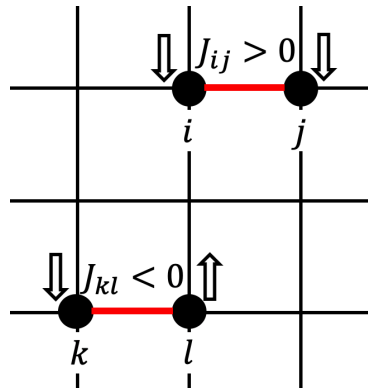


図 2.1 4 変数イジングモデルの例

アン全体の基底状態が一致しないことをフラストレーションと呼ぶ。フラストレーションがあるハミルトニアン基底状態を求めることは容易ではない。任意の組み合わせ最適化問題は、イジングモデルのハミルトニアン式 (2.1) として定式化可能 [8] であり、その解は基底状態として求めることができる。組合せ最適化問題は、ある制約条件のもとで、コスト関数を最大もしくは最小の値にする変数の組み合わせを求める問題であり、イジングモデルで定式化すると、多くの場合そのハミルトニアンはフラストレーションを持つ。計算量を変数の数に対して指数関数的に増加するため、従来、分枝限定法などの厳密解法だけではなくヒューリスティクスなど様々なアルゴリズムが研究されてきた。さらに、イジングモデルは、古典ニューラルネットワークにおいてはホップフィールドモデル、ボルツマンマシンとして導入されたモデルでもあり、機械学習分野においても重要な技術である [9]。

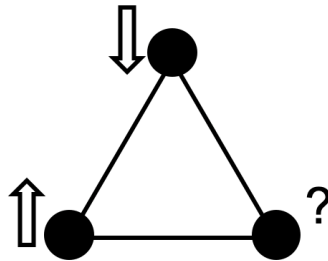


図 2.2 3 変数イジングモデルの例

QA は量子効果を利用してイジングモデルの基底状態を求める技術である。量子の重ね合わせの状態は次のパウリ行列と呼ばれる行列を利用して表現される。

$$\hat{\sigma}^x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}^z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.2)$$

QA は、パウリ行列を利用した横磁場イジングモデルにより探索を行う。

$$\hat{H} = - \sum_{i < j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z - \Gamma \sum_i \hat{\sigma}_i^x \quad (2.3)$$

この式の第 2 項目の Γ を非常に大きな値 (第 1 項目を無視できるほど大きな値) から 0 まですぐに変化させることで状態間の遷移を引き起こし、基底状態の探索が行われる。QA と同様にエネルギーの基底状態を探索することで解を得るヒューリスティックアルゴリズムにシミュレーテッド・アニーリング (Simulated Annealing; SA)[10] がある。エネルギーの基底探索において、SA が熱ゆらぎにより確率的な遷移をするのに対して、QA は量子ゆらぎにより状態遷移をする。SA と QA は十分長い時間をかけて系の温度を下げていくことで最適解に到達することが保証されている。しかし、一般的に SA が必要とする時間は非現実的なほど長く、計算時間を短くすることで最適解ではない解 (局所解) しか得られないことも少なくない。一方、QA は量子の性質により SA よりも局所解に陥りにくく、高速に (現実的な時間で) 最適解に到達できる可能性がある [5]。

また、式 (2.1) のハミルトニアンは、

$$x_i = \frac{\sigma_i + 1}{2} \quad (2.4)$$

とすることで、二値変数 $x_i \in \{0, 1\}$ を用いた二次制約なし二値最適化 (Quadratic Unconstrained Binary Optimization; QUBO) に変換することができる。

$$H(x) = \sum_i \sum_j Q_{ij} x_i x_j \quad (2.5)$$

ここで、 Q_{ij} は x_i と x_j 間の相互作用である。組合せ最適化問題の定式化ではイジングモデルよりも QUBO の方が扱いやすい場合が多いことから、本論文では QUBO を利用して議論を進める。

2.2 量子アニーリングにおける現状の課題

量子ビットを作る方法はイオントラップや光などいくつか研究されており、2011 年には D-Wave 社から超伝導集積回路を用いた QA の商用マシン [11, 12] の提供が開始された。その後、2019 年に Google[13] が特定の問題において量子コンピュータがスーパーコンピュータよりもはるかに高速に解くことができるという量子超越性を証明し、大きな注目を集めた。量子アニーリングマシンは、現在よりも大きなシステムへの開発が進められたとしても消費電力は基本的にほとんど増えないとされている [2]。しかし、量子ビットの接続を複雑にすることや、量子ビット数を大規模にすることは簡単ではない。そのため、量子インスパイア型アニーリング (QIA) マシン (イジングマシン, Simulated quantum annealing (SQA) マシンとも呼ばれる) も各社から提供されている [7, 14, 15]。QIA マシンは、超伝導集積回路を用いた商用マシンよりも安価に多くの量子ビットを扱うことができるため、QA の適用手法の研究やビジネスの場面では実用的な問題への活用利用され始めている。現在提供されている QA/QIA マシンとそれぞれの特徴は表 2.1 の通りである。本研究では、後述する「現状の QA/QIA マシンの利用における制約 (a)-量子ビット接続の制約」の影響がない

Fixstars Amplify Annealing Engine (Amplify AE) を利用した。

表 2.1 現状のアニーリングマシン [14]

	D-Wave 2000Q / Advantage	Fixstars Amplify AE	日立 CMOS アニーリングマシン	富士通 Digital Annealer (DA)	東芝 SQBM+
計算方式	QA	GPU	デジタル回路	デジタル回路	GPU
量子ビット数	2,048 / 5,760	262,144 以上	61,952	8,192 / 100,000	100,000
結合グラフ	キメラグラフ / ペガサスグラフ	全結合	キンググラフ	全結合	全結合
全結合換算ビット	64 / 124	131,072	176	8,192 / 100,000	31,000

現状の QA/QIA マシンの利用には、ハードウェアとソフトウェアのそれぞれの側面から制約がある。

(a) ハードウェアにおける制約

QA/QIA マシン (表 2.1) ごとに異なるハードウェア上の制約がある。

- 量子ビット数の制約

利用できる量子ビット数に限りがあり、解くことのできる問題サイズに限界がある。この課題に対して、量子ビット数が少なくなるような工夫 [16, 17, 18, 19] や、大きな問題を解くための問題分割手法 [20, 21] などの研究が進められている。

- 量子ビット接続の制約

量子ビット間の接続は接続数が多いほど高コストになる。Amplify AE や Fujitsu DA などの一部の QIA マシンは量子ビット間を全結合に接続しているが、D-Wave の QA マシンを始め、そのほかの QIA マシンでは疎結合グラフを採用している。疎結合グラフでは、頂点の次数 (接続数) よりも大きな次数の頂点を持つ変数をグラフを埋め込む際には頂点分割 (Chaininig) により頂点次数の調整が必要になる。Chaininig とは、図 2.3 のように埋め込み先のグラフよりも次数が大きい頂点 v_i を複製して頂点集合 $\phi(v_i)$ とすることで、グラフ埋め込みを行う手法である。このとき、頂点集合 $\phi(v_i)$ に含まれる頂点全てが同じ値になるように頂点間の相互作用を適切に設定する。Chaininig 後のグラフのイジングモデル (もしくは QUBO) の基底状態は、Chaininig 前のグラフのイジングモデル (もしくは QUBO) の基底状態と一致することが証明されている [22]。ただし、分割頂点数が増加すると、分割頂点間で異なる値になる Chain break が発生しやすくなり、最適解が得られにくくなることがある。さらに、Chaininig により計算に必要な量子ビット数が多くなり、QA マシンの物理グラフに埋め込めなくなったり、問題が解きにくくなる問題も発生することがある。

(b) ソフトウェアにおける制約

- ハミルトニアン の 定式化

QA/QIA を用いて問題を解く際には、問題を 2 次のイジングモデル、もしくは、QUBO で定式化しなければならない。そのため、3 次以上の高次式や不等式の

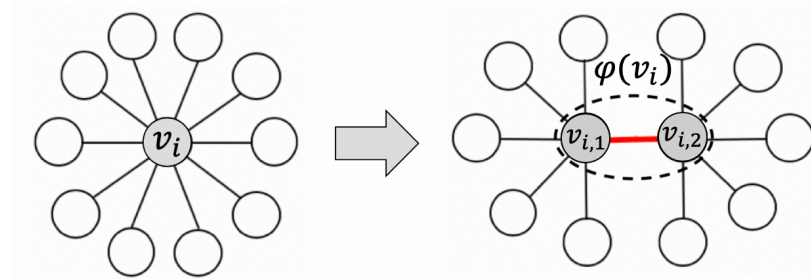


図 2.3 Chaininig[22]

場合は、補助変数などを導入して 2 次式に変換する必要がある。その変換により、変数の数が大幅に増加し、得られる解の精度も低下することがある。

- 目的関数の最小化

QA/QIA はイジングモデル、もしくは、QUBO で定式化されたハミルトニアンを最小化する。問題をイジングモデル（もしくは QUBO）に定式化する際には、目的と制約に分けて定式化し、それらに重み（ペナルティ）を付加して足し合わせることで QA で解くイジングモデル（もしくは QUBO）が生成される。この重みは、制約が満たされるように十分な大きさが必要であるが、大きすぎるとハミルトニアンにおける目的関数の影響が小さくなり最適解が得られにくくなるため、適切な重みの値を設定する必要がある。さらに、制約関数が増える程、問題が複雑になる、各制約関数に付加する重みの調整が難しくなるといった問題も発生する。その場合、制約を満たす解が得られにくくなったり、目的関数の最小化が十分に行われず、質の良い解が得られにくくなる。

現状の QA/QIA マシンを利用するうえで、これらの制約や課題に対処していかなければならない。これらのマシンを利用するための効果的な方法を開発することは、QA を実用的な問題に適用する上で重要な貢献となる。

第 3 章

疎行列の直接解法向けフィルイン削減オーダリング

この章では、疎行列連立一次方程式を直接解法で解く際に利用されるオーダリングについて説明する。3.1 節で疎行列のオーダリングについて説明し、3.2 節で従来手法を紹介する。

3.1 疎行列のオーダリング

科学技術計算において、疎行列連立一次方程式の求解は頻繁に登場する。このとき、疎行列の非零要素のみを格納することで、メモリの消費量と計算コストを削減する工夫がなされる。しかし、このような方程式を直接解法で解くと、疎行列の零要素が非零になるフィルインが頻繁に発生する。これにより、行列の疎性が損なわれ、メモリや計算のコストが増加してしまう。その問題に対して、あらかじめ行列の要素を並べ替えることでフィルインの数を減らすことができるオーダリング手法が長年に渡って研究されてきた [23, 24]。Yannakakis ら [25] は発生するフィルインの数を最小化する並べ替えを求めることは NP 完全問題であることを証明し、Formin ら [26] はそのようなオーダリングを厳密解法により行列サイズ n に対して $O(1.8899^n)$ 時間で求められることを示した。さらに、Rose ら [27] はフィルインが全く発生しないオーダリングである Perfect elimination ordering が存在する場合には、 $O(nm)$ 時間 (m は後述する消去グラフの辺の総数) で計算が可能であることを示した。Dahlhaus ら [28] は Perfect elimination ordering は存在しない問題でも最大次数が制限されたグラフに変換できる問題は $O(n(\Delta^3 + \alpha(n)))$ 時間で計算可能なことを示した。これらの厳密解法に関する研究に加え、短時間で計算可能であり、フィルイン数を削減する様々なヒューリスティックアルゴリズムも研究されている (詳細は 3.2 に後述)。

特に、対角要素に零要素を持たない対称行列は図 3.1 のような消去グラフ [24] と呼ばれるグラフで表現することができる。消去グラフは、行/列番号を頂点として表現することで、頂点の消去を前進消去などにおける行/列の消去として表現できる。また、グラフの辺は対称行列の非零パターンを表す。例えば、辺 $\langle u, v \rangle$ は u 行 v 列目と v 行 u 列目の要素が非零であることを表す。そのため、前進消去などにおける行/列の消去の際に生じるフィルイン

は、消去対象の頂点の任意の近傍頂点間に辺が存在するように追加される辺で表すことが可能である。例えば図 3.1 では、図左から中央への遷移で行列の 1 行目が消去される際に 2 行 3 列目と 3 行 2 列目の要素が非零になる。これを消去グラフでは、頂点 1 を消去し、近傍頂点の 2 と 3 を接続する辺が追加することで表現できる。また、図中央から図右への遷移で行列の 2 行目が消去される際に 3 行 4 列目と 4 行 3 列目、4 行 5 列目と 5 行 4 列目の要素が非零になる。これを消去グラフでは、頂点 2 を消去し、近傍頂点の 3 と 4、4 と 5 を接続する辺が追加することで表現できる。このステップで消去グラフは全結合となり、これ以上フィルインが発生しないことがわかる。この消去グラフは従来の様々なオーダリング手法で利用されている。消去グラフを利用した従来のヒューリスティックアルゴリズムについて次節で述べる。

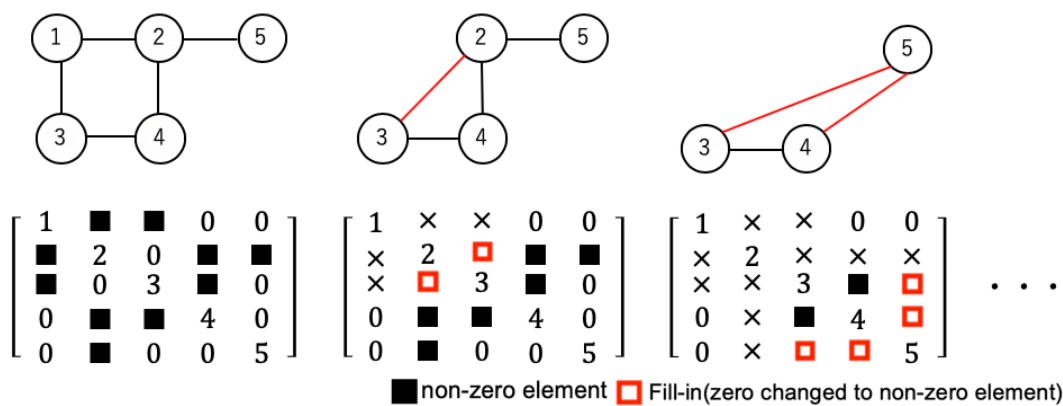


図 3.1 消去グラフの例

3.2 先行研究

従来のヒューリスティックアルゴリズムは、フィルインを削減するもの、並列性を向上させるものなどそれぞれ効果が異なる。本節では、消去グラフを利用してフィルインを削減するオーダリング手法を重点的に紹介する。

まず、最も一般的に利用されているアルゴリズムとして、Tinney ら [29] によって提案された Minimum degree (MD) 法がある。この手法では、各ステップで最小次数、つまり、頂点に接続している辺の総数が最小である頂点を 1 つずつ消去していく。この頂点の消去順がオーダリングとなる。このとき、最小次数頂点が複数ある場合は、辞書順やランダムなど任意に選択されることが多い。図 3.2 に 7×7 対称行列に対する MD の適用例を示す。この図 3.2 は、説明の都合上 MD によるオーダリング結果とそのまま前進消去した場合の結果が同じになる行列を利用している。まず、最小次数の頂点である頂点 1 を消去する。このとき、頂点 1 の近傍頂点である頂点 2 と頂点 7 が完全グラフとなるように、この 2 つの頂点を接続する辺を追加する。次のステップでは、最小次数となる頂点は頂点 2, 3, 5, 6 の 4 つである。ここでは、最も頂点番号が小さい頂点 2 を消去し、近傍頂点間が全結合になるように、頂点 3 と頂点 7 を接続する辺を追加する。残りのステップも同様に進めていくと、

最終的に 3×2 個のフィルインが発生することがわかる。この手法は、あまり大きくない問題では非常に効果的であり、特に、環状閉路を持たないグラフにはフィルインを発生させない [30]。MD は多くの研究で改良されており、George ら [31] による商グラフを利用する手法、Amestoy ら [32] による実行時間を短縮するために頂点次数を近似する Approximate minimum degree (AMD) 法などが提案されている。ただし、AMD は MD と同数以上のフィルインが発生する。

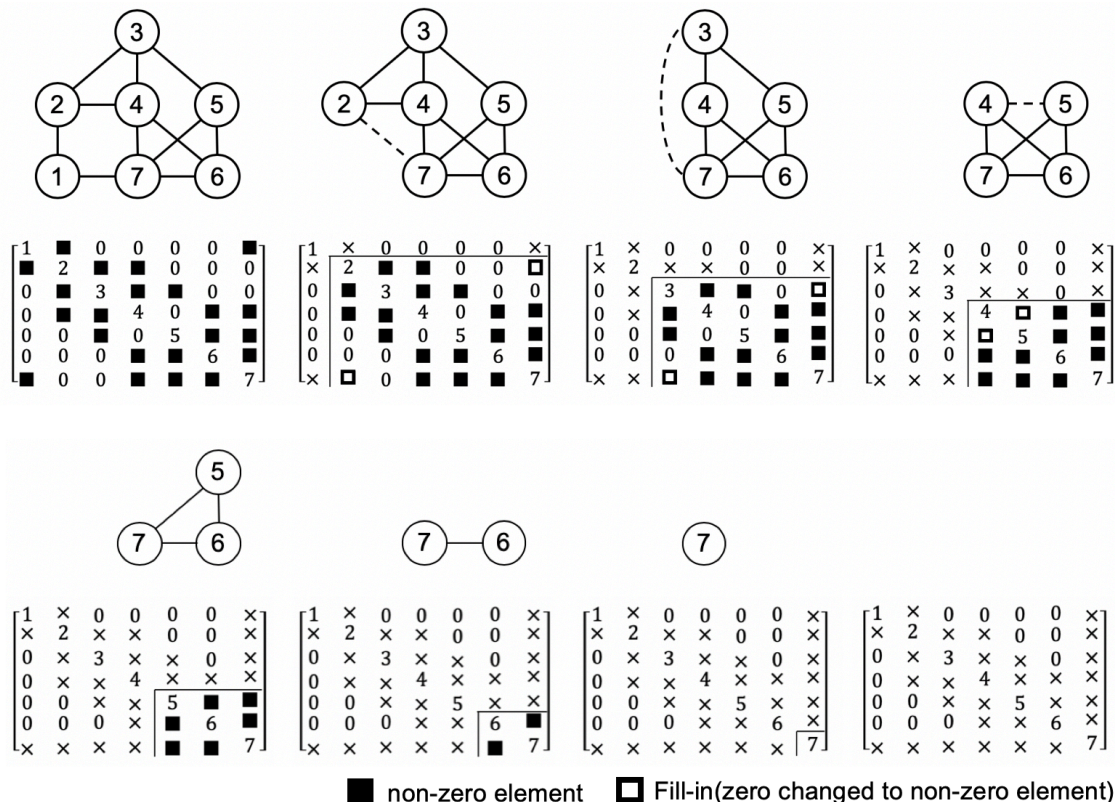


図 3.2 MD の適用例 [a]

もう 1 つのよく使われている手法として、Nested dissection (ND) 法がある。この手法は、George ら [33] により提案され、図 3.3 左のようにメッシュ構造の行列をレベルごとに再帰的に分割し、複数の小さな行列に分割することでオーダリングを行う。レベルごとに分割された行列は、図 3.3 中央のように隣接するレベルを接続することで 2 分木構造で表すことができ、図 3.3 右のように隣接するレベルに属す頂点間は全結合となるように辺が追加される (フィルインが発生する)。さらに、George ら [34] により、メッシュ構造のグラフだけでなく一般的なグラフでも動作するように改良された。ND ではフィルイン数を削減するだけでなく、分割した複数の行列を並列処理することも可能である。

さらに、Cuthill と McKee [35] によって提案された Cuthill-McKee (CM) 法は、フィルイン数だけでなく、行列のバンド幅やプロフィールも削減するため、行列計算を並列処理する際の並列性を向上させるためにも利用される。CM は、消去グラフを図 3.4 のように並列処理をする頂点ごとにレベル分けし、レベルごとにオーダリングの順番を割り当てる手法であ

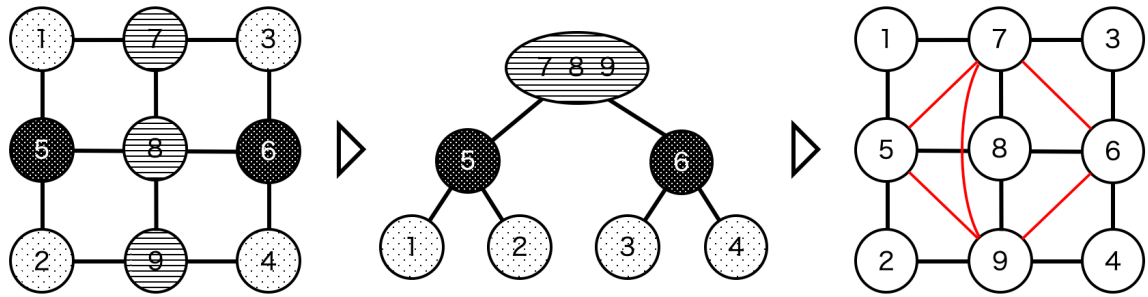


図 3.3 ND の適用例

る. CM と逆順に番号づけをする Reverse Cuthill-McKee (RCM) 法も George ら [36] によって提案されている. 一般的に, RCM の方が CM よりもフィルインが少なくなる傾向がある [37, 38]. 並列処理向け CM と RCM の手順を Algorithm 1 に示す. CM/RCM には, 開始頂点 (図 3.4 では左下の黒で塗りつぶされた頂点) から遠い頂点がレベル付けされるまでに多くのレベルが必要になる, 逐次処理のため頂点数が多くなるとレベル付けに時間がかかるといった課題がある. これらの課題を解消する手法として, RCM を適用した後に関連のないレベルに属する頂点を同じレベルに再度割り当てることでレベル数を削減する RCM+Cyclic multi coloring 法 [40] や, 開始頂点から距離が遠い点を同じレベルに設定し, それらの頂点から同時にレベル付けをすることでレベル数とレベル付けにかかる時間を削減する MIP-RCM 法 [37] などの改良法も提案されている.

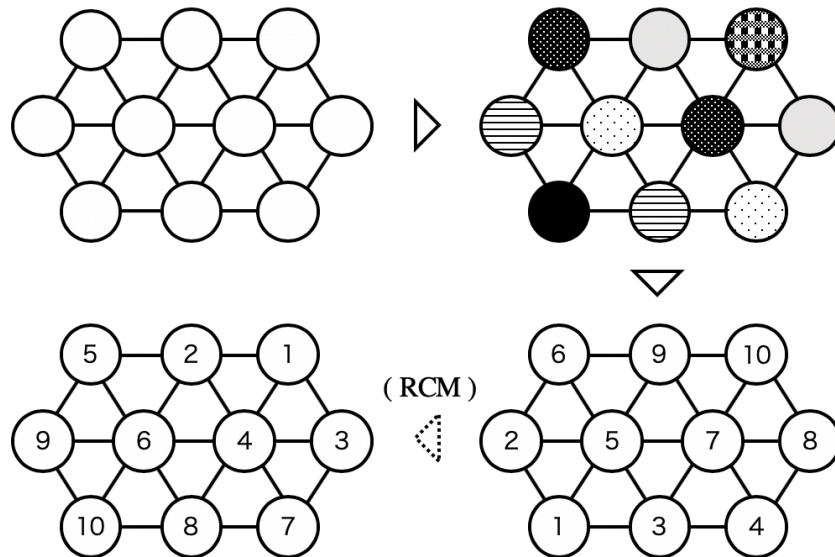


図 3.4 並列処理向け CM・RCM の適用例

Algorithm 1 並列処理向け CM・RCM [37, 39]

```
1:  $k = 1$ 
2: 最小次数の頂点のレベル =  $k$ 
3: while レベル付けされていない頂点が存在する do
4:    $k = k + 1$ 
5:    $k - 1$  レベルの頂点に隣接し,  $k$  レベルの頂点に隣接しない頂点のレベル =  $k$ 
6: end while
7:  $k = 1$ 
8:  $i = 1$ 
9: while 番号付けされていないレベルが存在する do
10:  while レベル  $k$  に属す頂点が存在する do
11:    if 頂点のレベル =  $k$  then
12:      頂点番号 =  $i$ 
13:       $i = i + 1$ 
14:    end if
15:  end while
16:   $k = k + 1$ 
17: end while
18: (RCM の場合) 頂点番号を逆順に付け直す
```

これらのアルゴリズムはそれぞれ利点や欠点が異なるため、組み合わせてハイブリッド手法として利用されることもある [30]. 本論文ではフィルイン削減のオーダリング手法の従来法として、今回紹介したオーダリング手法の中ではフィルイン数が最小になる MD を比較対象に用いる.

第 4 章

Quantum minimum fill-in algorithm

Quantum Minimum Fill-in (QMF) algorithm は、正定値対称行列に対するフィルイン数が最小になるオーダリングを求める問題を 0-1 整数最適化問題として定式化し、QA で解を求める手法である。フィルイン数削減オーダリングは従来多段的な手法を利用して求められてきたが、この手法は静的にオーダリングを求めることができる。QMF は、消去グラフを利用し、辺の追加が最小となる頂点の消去順を求める。この章の内容は関連論文 [a, b, c, d] を基に一部発展させたものである。

4.1 QUBO 定式化

頂点の消去順を表現する二値変数 $x_{n,s} \in \{0,1\}$ と、頂点 u と v 間の辺の存在を表現する二値変数 $y_{\langle u,v \rangle} \in \{0,1\}$ を用意する。 $x_{n,s} = 1$ は、頂点 n を第 s ステップに消去することを表し、 $y_{\langle u,v \rangle} = 1$ は、頂点 u と頂点 v の間の無向辺が存在する（非存在辺であれば追加される）、つまり、分解後の行列において (u,v) 要素または (v,u) 要素が非零であることを表す。この変数設定の例を図 4.1 に示す。図左は行列の消去グラフ、図中央は頂点 0 ~ 4 の消去ステップを表す変数集合 \mathbf{x} 、図右は各頂点間辺の有無を表す変数集合 \mathbf{y} である。また、この図では白抜きの変数は 0、黒で塗りつぶした変数は 1 を表す。QMF の QUBO 定式化の

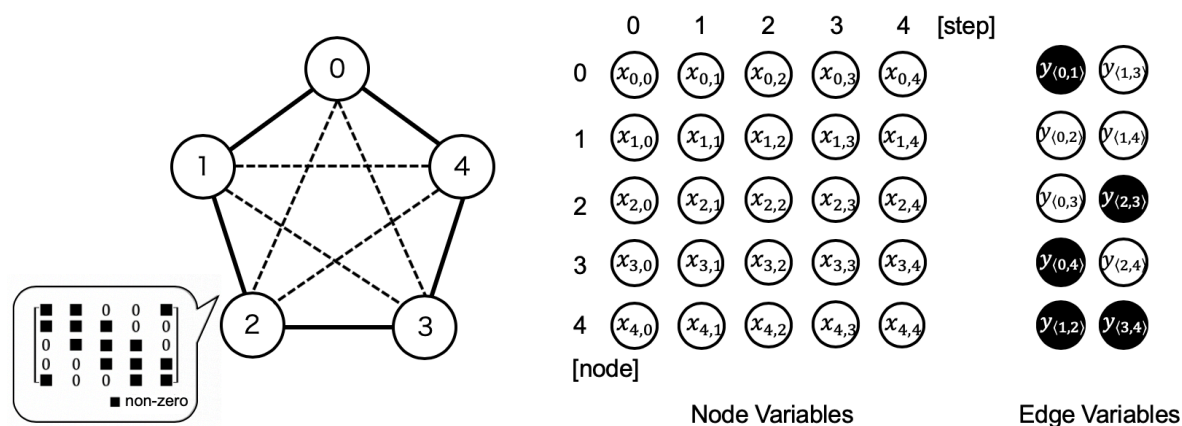


図 4.1 QMF の変数設定例

ために、消去グラフの頂点数 N とオーダリング実施前の消去グラフにおいて存在しない辺集合 \bar{E} を定義する。

まず、QMF の目的関数を定式化する。

目的関数. QMF は追加される辺の総数を最小化する

図 4.1 では、 $\bar{E} = \{\langle 0, 2 \rangle, \langle 0, 3 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 4 \rangle\}$ である。これを用いて目的関数は次式で定義される。

$$H_{cost} = \sum_{\langle u,v \rangle \in \bar{E}} y_{\langle u,v \rangle} \quad (4.1)$$

この H_{cost} は、辺の追加が発生しない、つまり、フィルインが発生しないとき最小値 0 をとる。

また、QMF には次の 2 つの制約がある。

制約 1. 全ての頂点が各ステップに 1 頂点ずつ消去される

この制約を満たすとき最小値 0 となる関数を次式で定義する。

$$H_1 = \sum_{n=0}^{N-1} \left(\sum_{s=0}^{N-1} x_{n,s} - 1 \right)^2 + \sum_{s=0}^{N-1} \left(\sum_{n=0}^{N-1} x_{n,s} - 1 \right)^2 \quad (4.2)$$

この関数の第 1 項目は図 4.2(a) のように、頂点 n を表す変数 $x_{n,s}$ のステップ方向の総和が 1 になるとき、つまり、頂点 n が 1 度だけ消去されるとき最小値 0 をとる。第 2 項目は図 4.2(b) のように、消去ステップ s を表す変数 $x_{n,s}$ の頂点方向の総和が 1 になるとき、つまり、各消去ステップ s において頂点が 1 つだけ消去されるとき最小値 0 をとる。

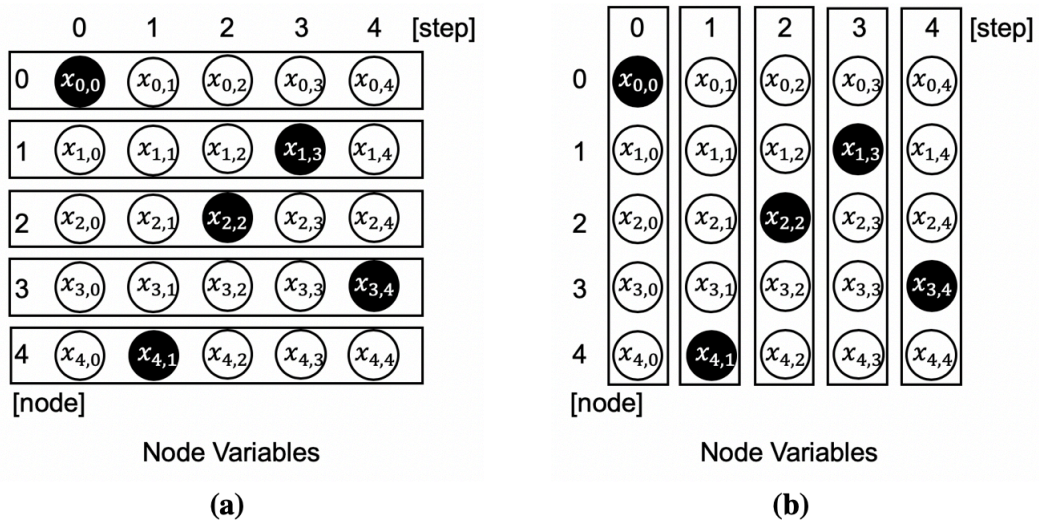


図 4.2 制約 1 の変数値例

制約 2. 頂点の消去により必要な辺が追加される

各ステップにおいて、図 4.3 のように消去対象の頂点に接続のある頂点間が消去後に全結合となるように辺の追加を行う。ここで辺の追加は次の条件により発生する。

- 頂点 n を s ステップ目で消去する ($x_{n,s} = 1$)
- 頂点 u が s ステップ目よりも前のステップで消去されていない ($\sum_{k=0}^{s-1} x_{u,k} = 0$)
- 頂点 v が s ステップ目よりも前のステップで消去されていない ($\sum_{k=0}^{s-1} x_{v,k} = 0$)
- 消去する頂点 n と頂点 u を結ぶ辺が存在する ($y_{\langle n,u \rangle} = 1$)
- 消去する頂点 n と頂点 v を結ぶ辺が存在する ($y_{\langle n,v \rangle} = 1$)

以上の条件が全て満たされた時に、頂点 u と頂点 v を接続する辺を追加する ($y_{\langle u,v \rangle} = 1$)。

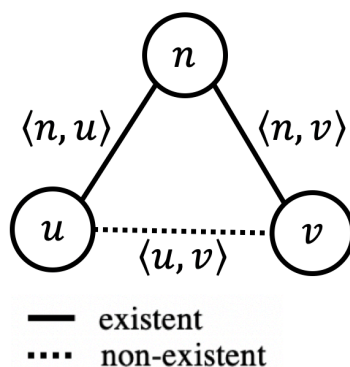


図 4.3 制約 2 の辺と頂点

以上の複雑な制約を表現する QUBO を定式化することは簡単ではない。例えば、図 4.4(a)~(c) の頂点数が異なる 3 つの消去グラフを考える。この図にはそれぞれのグラフを QMF で解いたときの取り得るエネルギー値のイメージも示しており、制約を満たす解のエネルギー値を黒丸、制約違反の解を白丸で示している。図 4.4(a) のように 3 頂点のみの局所的なエネルギー値だけを考えると次の QUBO が定式化できる。

$$h = x_{n,s}(1 - y_{\langle u,v \rangle}) + y_{\langle n,u \rangle}y_{\langle n,v \rangle} + \left(1 - \sum_{k=0}^{s-1} x_{u,k}\right) \left(1 - \sum_{k=0}^{s-1} x_{v,k}\right) \quad (4.3)$$

この関数は、3 頂点のみの局所的なエネルギー値だけをみると、前述の制約を満たすときに最小値 0、それ以外ときに正の値となる。しかし、図 4.4(b), (c) のように非存在辺が多くなるにつれて制約を満たす解のエネルギー値と制約を満たさない解のエネルギー値が二極化しなくなる。このように、複雑な制約を持つ問題を 2 次式で定式化することは難しい。

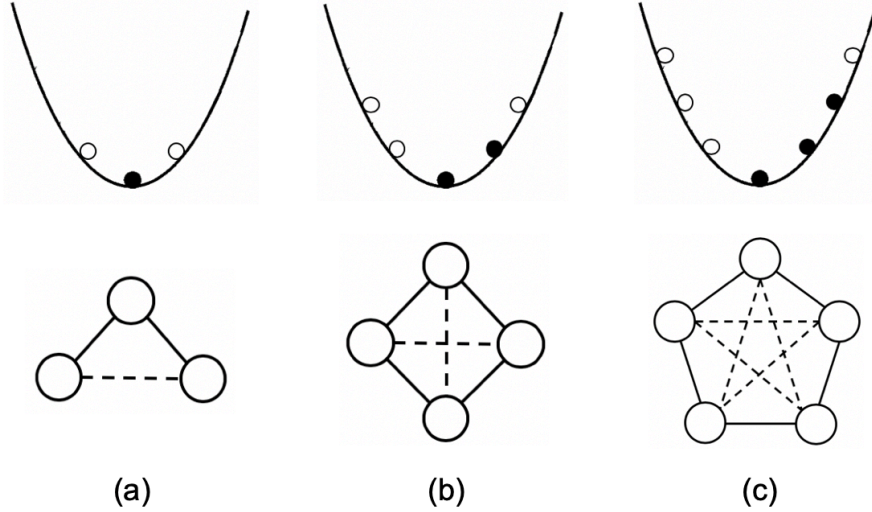


図 4.4 式 (4.3) を利用した場合のハミルトニアン全体のエネルギー値

そこで、制約 2 は各条件の積の形式で定義した。

$$H_2 = \sum_{(u,v) \in \bar{E}} \sum_{n \notin u,v} \sum_{s=0}^{N-2} (1 - y_{\langle u,v \rangle}) x_{n,s} \left(1 - \sum_{k=0}^s x_{u,k} \right) \left(1 - \sum_{k=0}^s x_{v,k} \right) y_{\langle n,u \rangle} y_{\langle n,v \rangle} \quad (4.4)$$

この関数は 6 次式であるため、QUBO として定式化するために 2 次式に変換する必要がある。この変換手法は 4.2 節と 4.3 節で議論する。

実用的な問題は制約が複雑な場合が多く、定式化すると 3 次以上の高次式になることがある。本節で定式化した QMF の制約関数 (4.4) もその例の一つである。このような 3 次以上の項を持つ問題は高次二値最適化 (High-order binary optimization; HOBO) と呼ばれ、補助変数を導入することで QUBO に変換することができる。しかし、一般的にこの変換には多くの補助変数が導入され、問題規模が急激に大きくなることから問題を解きにくくする。そこで、4.2 節と 4.3 節では HOBO が QUBO に変換される際に導入される補助変数の数を削減する工夫について議論する。

4.2 高次式の次数削減

高次式の次数削減とは、高次多項式に対して置換や式変形を適用することで低次の多項式に変換することである。本節は、特に、関連文献 [a] を基に一部発展させたものである。次数削減手法として様々な方法が提案されているが、本論文では Amplify AE に実装されている 3 つの変換手法を紹介する。

- KZFD[41, 42]

x_i ($i = 1, 2, 3$), a をバイナリ変数として、 $-x_1x_2x_3$ と $-a(x_1 + x_2 + x_3 - 2)$ はどちらも $x_1 = x_2 = x_3 = 1$ で最小値 -1 を与える。これを高次に拡張すると、 $-\prod_{i=1}^d x_i$

と同じ値を与える x_i および a について 2 次以下の多項式を求めることができる。したがって、 d 次のバイナリ変数多項式の負の項に対して、補助変数 a を 1 つだけ導入することで次の変換が可能である。

$$-\prod_{i=1}^d x_i \rightarrow (d-1)a - a \sum_{i=1}^d x_i \quad (4.5)$$

- **Ishikawa[43]**

KZFD 法による次数削減では、対象とする高次項は負である必要がある。Ishikawa 法は、KZFD 法を拡張することで、この制約を取り除いた。対称多項式を用いて、 d 次の正の項を $\lfloor (d-1)/2 \rfloor$ 個の補助変数 a_{n_d} を用いて次の変換を行う。

$$\begin{aligned} \prod_{i=1}^d x_i &\rightarrow \sum_{i=1}^{n_d} a_i (c_{i,d}(-S_1 + 2i) - 1) + S_2 \\ n_d &= \left\lfloor \frac{(d-1)}{2} \right\rfloor, \quad S_1 = \sum_{i=1}^d x_i, \quad S_2 = \sum_{i=1}^{d-1} \sum_{j=i+1}^d x_i x_j \\ c_{i,d} &= \begin{cases} 1 & (\text{if } d \text{ is odd and } i = n_d) \\ 2 & (\text{otherwise}) \end{cases} \end{aligned} \quad (4.6)$$

以降では、高次項が正、負どちらの場合も区別せず、Ishikawa 法と参照する。

- **SUBSTITUTION[14, 44]**

バイナリ変数のペア (x_i, x_j) に対して、積 $x_i x_j$ をバイナリ補助変数 a_{ij} で置き換える手法である。 $x_i x_j = a_{ij}$ が成り立つとき 0、それ以外るとき正の値を与える関数 g が制約として追加される。

$$\begin{cases} x_i x_j \rightarrow a_{ij} \\ g(x_i, x_j, a_{ij}) = x_i x_j - 2x_i a_{ij} - 2x_j a_{ij} + 3a_{ij} \end{cases} \quad (4.7)$$

4 次以上の項に対しては、この置換を再帰的に行うため、次数 n の項に対して、 $n-2$ 個の補助変数が必要になる。ただし、この置換を全ての項に対して同時に適用することで補助変数を共有できることがある。

ここから、QMF の高次式に次数削減を適用する方法を示す。QMF の制約関数 (4.4) は展開すると 3~6 次の項を含む多項式となる。これらの項をそれぞれ次数削減すると、Ishikawa 法で $12N(N-1)\bar{E}$ 、SUBSTITUTION 法で $6N(N-1)\bar{E}$ もの補助変数の追加が必要になる。このように多くの補助変数が必要になる原因の一つは、制約関数 (4.4) は多項式であり、それぞれの項を次数削減する必要があるためである。そこで、制約関数 (4.4) が

単項式になるように、次の補助変数 \mathbf{a} を導入する.

$$a_{\langle u,v \rangle} = 1 - y_{\langle u,v \rangle}, \quad a_{u,s} = 1 - \sum_{k=0}^{s-1} x_{u,k}, \quad a_{v,s} = 1 - \sum_{k=0}^{s-1} x_{v,k} \quad (4.8)$$

これらの等式は次の QUBO で定義される.

$$H_{2,\mathbf{a}}^1 = \sum_{\langle u,v \rangle \in \bar{E}} (a_{\langle u,v \rangle} + y_{\langle u,v \rangle} - 1)^2 + \sum_{s=0}^{N-1} \left(a_{u,s} + \sum_{k=0}^{s-1} x_{u,k} - 1 \right)^2 + \sum_{s=0}^{N-1} \left(a_{v,s} + \sum_{k=0}^{s-1} x_{v,k} - 1 \right)^2 \quad (4.9)$$

補助変数 \mathbf{a} を利用した制約関数 (4.4) は次式である.

$$H_{2,\mathbf{a}}^2 = \sum_{\langle u,v \rangle \in \bar{E}} \sum_{n \notin i,j} \sum_{s=0}^{N-1} a_{\langle u,v \rangle} x_{n,s} a_{u,s} a_{v,s} y_{\langle n,u \rangle} y_{\langle n,v \rangle} \quad (4.10)$$

この関数を Ishikawa 法で次数削減すると、 $2N(N-1)\bar{E}$ の補助変数で次数削減可能であり、補助変数 \mathbf{a} の分も加えても $5N(N-1)\bar{E}$ 程度の補助変数で QUBO に変換できる. ただし、疎行列の場合、存在しない辺の総数は $O(N^2)$ であるため、式 (4.10) の変数の総数は $O(N^4)$ である.

次数削減手法毎の変数の数を比較した結果を示す. 対角要素が非零であり、その他の要素が $1/3$ の確率で非零となるようにランダム生成した 200 個の $N \times N$ 行列における計算に必要な変数の数を示す. 次数削減前の関数に利用されている変数の数の平均 (input), (1) 補助変数 \mathbf{a} を導入した制約関数 (4.10) に Ishikawa 法を適用した場合, (2) 元の制約関数 (4.4) に Ishikawa 法と (3) SUBSTITUTION 法をそれぞれ適用して次数削減した場合にそれぞれ必要になる変数の数の平均 (# variables), (1) の変数の数に対する変数の増加率 ($/$ (1)) を表 4.1 に示す. 表 4.1 から, (1) 補助変数 \mathbf{a} を導入した制約関数 (4.10) を Ishikawa 法により次数削減したときと比較して, (2) 元の制約関数 (4.4) を Ishikawa 法により次数削減したときは 9.0 ~ 61.2 倍, (3) 元の制約関数 (4.4) を SUBSTITUTION 法により次数削減したときは 1.6 ~ 3.2 倍もの変数が必要になる. 補助変数 \mathbf{a} を導入するわずかな式変形の工夫により, 大幅に補助変数の増加が抑えられる.

4.3 不等式の利用

4.3.1 概要

本節は、特に、関連文献 [d] を基にしたものである. 複雑な制約をもつ問題 (主制約) を不等式を用いた QUBO として定式化する. 主制約の複雑な条件をそれぞれ副制約 $\bar{p}_i (i = 1$

表 4.1 次数削減後の変数の総数

N	input	(a) (4.10) + Ishikawa	(b) (4.4) + Ishikawa		(c) (4.4) + SUBSTITUTION	
	# variables	# variables	# variables	/ (a)	#Variables	/ (a)
5	32	515	4658	9.0	844	1.6
6	46	1172	14241	12.1	2062	1.8
7	63	2437	38722	15.9	4610	1.9
8	83	4515	90541	20.1	9223	2.0
9	105	7492	183700	24.5	16228	2.2
10	130	11894	351941	29.6	27802	2.3
11	158	17933	627058	35.0	44770	2.5
12	188	26017	1061581	40.8	69037	2.7
13	221	36247	1702895	47.0	101786	2.8
14	256	49620	2670391	53.8	148734	3.0
15	295	67137	4105577	61.2	214411	3.2

, 2, ..., N) に分けて考える. このとき, 主制約は全ての副制約 \bar{p}_i が違反したときに違反する. まず, 副制約 \bar{p}_i に対して以下の一次関数 \bar{P}_i を定義する.

$$\bar{P}_i = \begin{cases} 0 & \text{(when the constraint } \bar{p}_i \text{ is satisfied),} \\ 1 & \text{(when the constraint } \bar{p}_i \text{ is not satisfied).} \end{cases}$$

主制約を満たす関数は, 次の不等式で表すことができる.

$$\bar{P}_1 + \bar{P}_2 + \dots + \bar{P}_N \leq N - 1 \quad (4.11)$$

ただし, 次のような場合はこの方法を適用できない.

- 副制約 \bar{p}_i が線形関数として定義できない場合 ($(ax^2 + by^2)^n$ など)
- 最小値 (最大値) が 0 にならない関数 \bar{P}_i となる場合

例えば, 最小彩色問題 [45] は, グラフの頂点の彩色数を最小化するコスト関数を持つ.

$$H_{cost} = - \sum_{c=0}^{|V|-1} \prod_{v \in V} (1 - x_{v,c}) \quad (4.12)$$

ここで, V はグラフの頂点集合, $|V|$ は V の総数である. 式 (4.11) は, このコスト関数のように最小値 (最大値) が 0 ではない制約を定式化することはできない.

- 単項式 (x^n や $(1-x)^n$ など) となる場合は次数削減を利用した方が補助変数の追加が少なくなる

QA は不等式をそのまま扱うことができない. そのため, 次式のように不等式にスラック

変数 $z = (z_0, z_1, \dots, z_{L-1})^\top$ を導入して QUBO に変換する必要がある.

$$X \leq L \implies \left(X - \sum_l z_l \right)^2 \quad (4.13)$$

この変換にはいくつかの手法があり, それぞれ導入されるスラック変数の数や, 数 X を表現できるスラック変数の組み合わせ数である表現力が異なる.

1. Unary encoding

ある数 X を L 個のスラック変数の和で表現する変換手法. 数 X を表現可能な z の組み合わせ (表現力) は, $L C_X$ 通りある.

$$\left(X - \sum_{l=0}^{L-1} y_l \right)^2 \quad (4.14)$$

2. Binary encoding

$\lceil \log_2 L + 1 \rceil$ 個のスラック変数を利用して, 2 進数で数 X を表現する手法. 数 X を表現可能な c の組み合わせは 1 通りである.

$$\left(L - X - \sum_{l=0}^{\lceil \log_2 L + 1 \rceil} 2^l z_l \right)^2 \quad (4.15)$$

3. Linear encoding

$\lceil \sqrt{L} \rceil$ 個のスラック変数を利用して, $0 \sim \lceil \sqrt{L} \rceil$ のうちのいくつかの数の和で数 X を表現する手法. 数 X を表現可能な z の組み合わせは 1 通り以上ある.

$$\left(L - X - \sum_{l=1}^{\lceil \sqrt{L} \rceil} l z_l \right)^2 \quad (4.16)$$

4. One-hot encoding

$L + 1$ 個のスラック変数 z のうち, 数 X を z_X を 1, それ以外を 0 とすることで表現する手法. 数 X を表現可能な z の組み合わせは 1 通りであり, 式 (4.18) の制約関数が追加される.

$$\left(X - \sum_{l=0}^L l z_l \right)^2 \quad (4.17)$$

$$\left(\sum z_l - 1 \right)^2 \quad (4.18)$$

また, 不等式を QA で扱うための手法として拡張ラグランジュ法 [46] やそれを拡張した

手法 [16] も利用されている。これらの手法は反復計算により解を求めるものであり、計算時間が反復回数に依存する。そのため、本論文では 1 反復で解を得るスラック変数による変換のみを議論する。

4.3.2 適用例

本節では、不等式を利用した定式化を「(weighted) maximum satisfiability 問題」と「QMF の制約 2」, 「K-Coloring 問題の One-hot 制約」に適用した例を示す。

(weighted) maximum satisfiability (MaxSAT) problem

MaxSAT 問題は、充足可能性問題を最適化問題に拡張したものであり、充足する節の数を最大にする変数の組合せを求める問題である [47, 48]。特に、重み付き MaxSAT 問題では、すべての節に非負の重みが与えられ、満足する節の重みの和を最大にする解を求める。MaxSAT は NP 困難問題であり、様々な解 [49] が研究されている。 $x \in \{0, 1\}$, $c = x_0 \vee \bar{x}_1 \vee x_2 \vee \bar{x}_3$ のとき, $S_c^+ = \{x_0, x_2\}$, $S_c^- = \{x_1, x_3\}$, 節の集合 C , 節 c に与えられる非負の重み w_c と定義すると、この問題は HOBO として定式化できる [48]。

$$H_{HOBO}^{MaxSAT} = \sum_{c \in C} w_c \prod_{u \in S_c^+} (1 - u) \prod_{v \in S_c^-} v \quad (4.19)$$

この式の w_c 以降の部分は、論理式が真のときは 0, 偽のときは 1 である。従って、この HOBO を最小化する解が MaxSAT の最適解となる。

一方、この問題は、節 c の長さ $|c|$ を用いて、次の不等式で表すこともできる。

$$\sum_{u \in S_c^+} (1 - u) + \sum_{v \in S_c^-} v \leq |c| - 1 \quad \forall c \in C \quad (4.20)$$

この不等式の左辺は、論理式 c が真のとき $0 \sim |c| - 1$ の値をとり、偽のとき $|c|$ をとる。つまり、この不等式が成り立つとき、論理式 c は真である。この不等式はスラック変数 z を導入することで、次式の QUBO で定義できる。

$$h_{QUBO}^{MaxSAT} = \left(\sum_{u \in S_c^+} (1 - u) + \sum_{v \in S_c^-} v - \sum_l^{l=|c|-1} z_l \right)^2 \quad \forall c \in C \quad (4.21)$$

この関数は、論理式 c が真のとき 0, 偽のとき 1 となる。この関数を用いて重み付き MaxSAT の最適解を求めることは、以下の QUBO を最小化する u と v の組み合わせ

せを求めることと等価である。

$$H_{QUBO}^{MaxSAT} = \sum_{c \in C} w_c \left(\sum_{u \in S_c^+} (1 - u) + \sum_{v \in S_c^-} v - \sum_l^{|c|-1} z_l \right)^2 \quad (4.22)$$

QMF (制約 2)

制約 2 を主制約, 制約 2 のそれぞれの条件を副制約とすると次の不等式で定式化できる。

$$(1 - y_{\langle u, v \rangle}) + x_{n,s} + \left(1 - \sum_{k=0}^s x_{u,k} \right) + \left(1 - \sum_{k=0}^s x_{v,k} \right) + y_{\langle n, u \rangle} + y_{\langle n, v \rangle} \leq 5 \quad (4.23)$$

この不等式の左辺は, 5つの副制約がすべて満たされるが辺 $y_{\langle u, v \rangle}$ が追加されないとき最大値 6 をとり, それ以外のときは 0 ~ 5 の値をとる。したがって, この不等式が成り立つとき, 主制約が満たされる。さらに, この不等式はスラック変数 z を導入することで, QUBO に変換することができる。

$$H_{2,b} = \sum_{\langle u, v \rangle \in \bar{E} \cap \{u, v\}} \sum_{s=0}^{N-1} \left\{ (1 - y_{\langle u, v \rangle}) + x_{n,s} + \left(1 - \sum_{k=0}^s x_{u,k} \right) + \left(1 - \sum_{k=0}^s x_{v,k} \right) + y_{\langle n, u \rangle} + y_{\langle n, v \rangle} - \sum_{l=0}^4 z_l \right\}^2 \quad (4.24)$$

この関数は, この制約が満たされていれば 0 をとり, そうでなければ 1 をとる。

K-Coloring 問題

K-Coloring 問題は, グラフの頂点集合 V , 辺集合 E としたとき, このグラフの全ての頂点を K 種類の色を用いて彩色する問題である。このとき, 各頂点は必ず 1 色で彩色され, 隣接する頂点は異なる色で彩色されるようにする。この問題は $|V|K$ 個の変数を用いて QUBO で定式化できる。

$$\sum_{v \in V} \left(1 - \sum_{k=0}^{K-1} x_{v,k} \right)^2 + \sum_{(u,v) \in E} \sum_{k=0}^{K-1} x_{u,k} x_{v,k} \quad (4.25)$$

この QUBO の第 1 項目の One-hot encoding による関数は, 必要な変数の数が多くなることが課題である。そこで, 2 進数を利用することで One-hot encoding よりも少ない変数で定式化可能な Binary encoding に変更することを考える。One-hot encoding では, 頂点 v を色 k で彩色するかを表す二値変数 $x_{v,k}$ を利用して定式化した, Binary encoding では, 頂点 v を色 $\sum_{i=0}^{K-1} 2^i b_v^i$ で彩色すること表す二値変数

b_v を利用する. Binary encoding を利用して式 (4.25) の第一項目を変換すると次式になる.

$$H_{QUBO} = \sum_{i=0}^{\log_2 K} \prod_{(u,v) \in E} (1 - b_u^i - b_v^i)^2 \quad (4.26)$$

この高次式を不等式を利用して定式化すると次のようになる.

$$\begin{aligned} (1 - b_u^0) + (1 - b_v^0) + (1 - b_u^1) + (1 - b_v^1) + \dots + b_u^i + b_v^i &\leq 2\log_2 K - 1 \\ (1 - b_u^0) + (1 - b_v^0) + \dots + b_u^{i-1} + b_v^{i-1} + (1 - b_u^i) + (1 - b_v^i) &\leq 2\log_2 K - 1 \\ &\vdots \\ b_u^0 + b_v^0 + (1 - b_u^1) + (1 - b_v^1) + \dots + (1 - b_u^i) + (1 - b_v^i) &\leq 2\log_2 K - 1 \end{aligned} \quad (4.27)$$

これを QUBO に定式化すると, $|V|\log_2 K + \log_2 K(2\log_2 K - 1)|E|$ 個の変数が必要になる. この QUBO で最適解は得られるが, One-hot encoding で定式化するよりも必要な変数と制約の数が大幅に増加する.

この他にも, Ide ら [50] の研究などの HOBO で定式化され, 前述の適用条件を満たす問題に適用可能である.

4.3.3 評価

QMF と MaxSAT を例に不等式を利用した定式化手法を評価する. この評価では, Amplify AE を求解にかける時間である annealing time を QMF は 2000ms, MaxSAT は 1ms に設定して利用した. QMF の問題行列は, 対角要素は非零であり, その他の要素は $1/3$ の確率で非零となるランダムに生成した 100 個の対称行列である. また, MaxSAT は, 要素数 N と節数 C が等しいランダムに生成した問題を利用する. さらに, 計算に必要な量子ビット数が増えるにつれて現在の QIA マシンでは実行可能解を得ることが難しくなるため, 変数の数の上限を 10,000 とした.

まず, 不等式による QUBO と HOBO を次数削減した QUBO の結果を比較する. 不等式による QUBO は, Unary encoding を利用した. QMF の結果を図 4.5 と図 4.6 に示す. まず, 図 4.5 は, 問題サイズに対する QUBO の変数の総数を示す. ここから, 不等式による QUBO は HOBO を次数削減した QUBO よりも変数の数が少ないことがわかる. 図 4.6 は QMF の 2 つの制約条件を満たす解の割合である実行可能解率を示しており, 特に, 黒塗りが最適解率, 灰色塗りが局所解率である. ただし, HOBO を Ishikawa 法により次数削減した QUBO では, 6×6 以上の行列では変数数が 10,000 個以上となるため実行可能解の計測は省略した. この結果から, 不等式による QUBO は HOBO を次数削減した QUBO よりも実行可能解率が高く, 最適解も多く得られていることがわかる.

MaxSAT の QUBO の変数の総数を図 4.7 に示す. この図 4.7 から, 不等式による QUBO

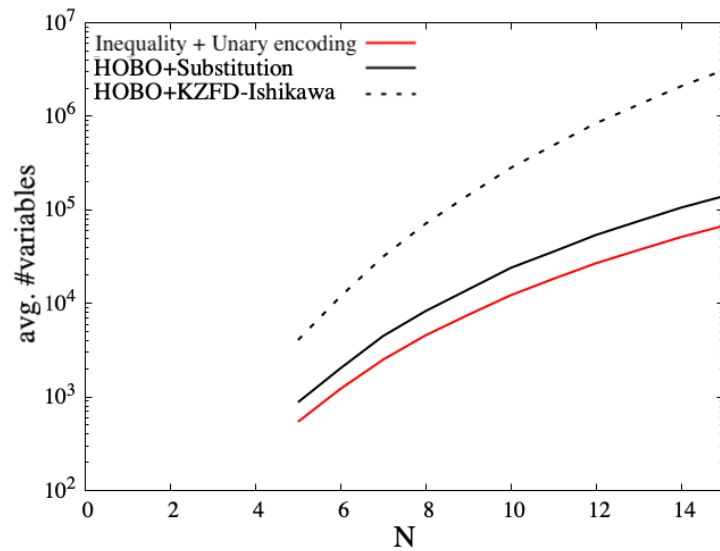


図 4.5 QMF の QUBO の変数の総数 [a]

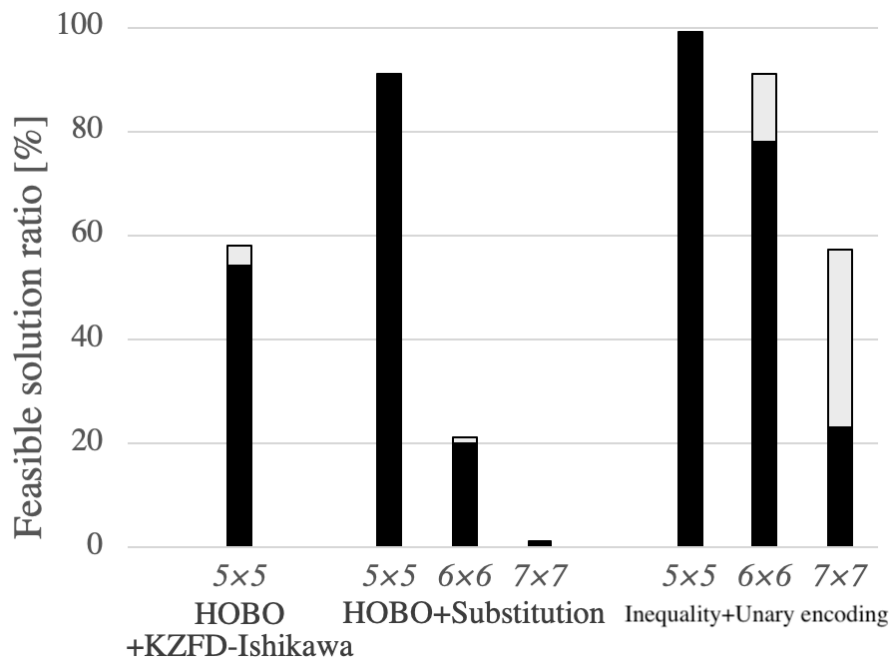


図 4.6 QMF の実行可能解数 [a]

は HOBO を次数削減した QUBO よりも変数の増加が小さいことがわかる。節数 $N < 9$ のときは、HOBO は 4 次以下の式のため次数削減によって追加される補助変数の数が少ない。そのため、このような小さな問題に不等式を用いるとかえって多くの補助変数が必要になる。 $N \leq 9$ のときは、不等式による QUBO は HOBO を次数削減したときよりも変数の数が少なくなる。得られた解のコスト関数の値は、提案手法と次数削減を用いた HOBO で同じ、つまり、得られた解の質は両手法とも同じであった。

以上の結果から、どちらの問題でも不等式を利用した QUBO の方が HOBO に次数削減をした QUBO よりも少ない変数で同等以上の質の解が得られることがわかる。したがっ

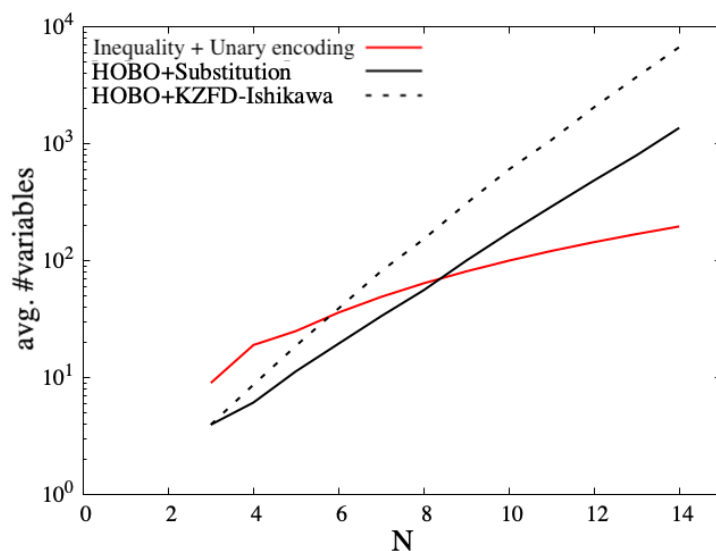


図 4.7 MaxSAT の QUBO の変数の総数 [a]

て、不等式を利用した定式化は限られた量子ビット数の QA/QIA マシンを効果的に利用するための有効な手法であると考えられる。本手法は、高次になってしまう制約のために QA の使用が避けられていた問題への QA の適用を可能にする。

次に、不等式の QUBO 変換手法の比較結果を示す。QMF の結果として図 4.8 に問題サイズに対する QUBO の変数の総数、図 4.9 に QMF の 2 つの制約条件を満たす解の割合である実行可能解率を示す。特に、図 4.9 では黒塗りが最適解率、灰色塗りが局所解率である図 4.8 から、Binary encoding, Linear encoding, Unary encoding, One-hot encoding の順に変数の数が少ないことがわかる。なお、Binary encoding, Linear encoding の変数の数は同じである。さらに、図 4.9 の結果から、実行可能解率と最適解率どちらも Unary encoding, Linear encoding, Binary encoding の順に高くなることがわかる。特に、Binary encoding と Linear encoding では、 7×7 行列に対して実行可能解がほとんど得られなかった。ここから、スラック変数の表現力が高いほど、質の良い解が得られやすいことが考えられる。ただし、One-hot encoding は実行可能解が得られなかったため、図 4.9 に結果を示していない。これは One-hot encoding では制約関数が追加されるため、実行可能解を得ることが難しくなるためと考えられる。以上の結果から、QMF のように複雑な制約を持つ問題には表現力の高い、つまり、解の数が増える Unary encoding が最も適していると考えられる。これは、QA や QIA がエネルギー曲線上の基底状態を探索するメタヒューリスティック解法であるため、解になり得る場所が多いほど解が得られやすいことが理由であると考えられる。田村ら [51] の研究でも、Binary 法よりも Unary の方が良解が得られ、Onehot 法で解が得られない理由として同様の考察がされている。

次に、MaxSAT に対する QUBO の変数の総数を図 4.10 に示す。この図 4.10 から、Binary encoding, Linear encoding, Unary encoding, One-hot encoding の順に変数の数が少ないことがわかる。得られた解のコスト関数の値、つまり、解の質はどの変換手法を用いても同じであった。ここから、MaxSAT のように制約がない、もしくは、少ないシンプル

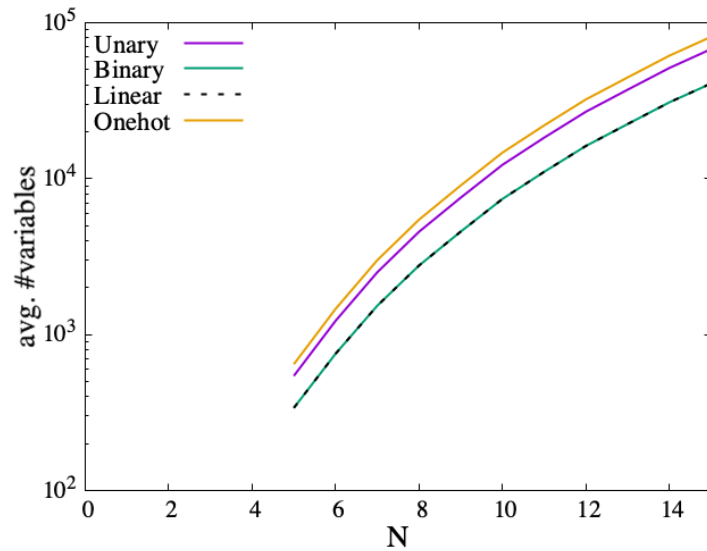


図 4.8 QMF の QUBO の変数の総数 [a]

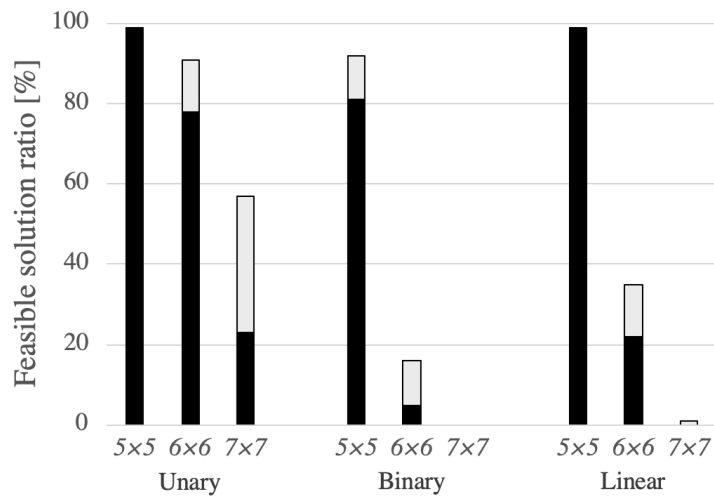


図 4.9 QMF の実行可能解数 [a]

な問題に対しては、変数の数が少ない Binary encoding が最も適していると考えられる。

以上の 2 つの問題の結果から、不等式の QUBO 変換手法は、問題の性質に応じて選択する必要があることがわかった。複雑な制約を持つ問題には表現力の高い変換手法、制約が少ない問題にはスラック変数の数が少ない変換手法が適していると考えられる。これらの結果は、不等式の QUBO 変換手法を選択するための新たな指針となる。その他の不等式の変換手法の比較結果は付録を参照されたい。

4.4 パラメータチューニング

この節では、QA や QIA の利用におけるパラメータチューニングの必要性について議論する。本節は、特に、関連文献 [6] を基にしたものである。QA や QIA を利用して解の探索を行う際には、次の 2 種類のパラメータの調整が必要になる。

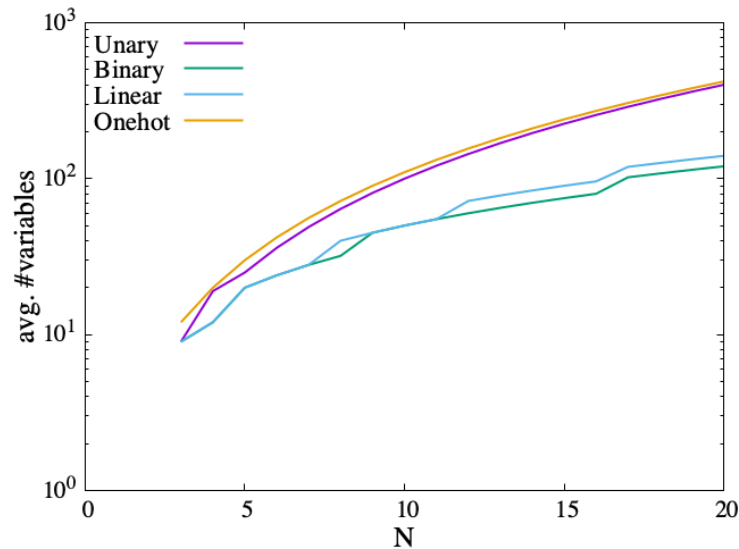


図 4.10 MaxSAT の QUBO の変数の総数 [a]

a. **ハミルトニアンのコストと制約関数の重み**

ハミルトニアンとして問題を定式化する際に、コストと制約関数の重みを与える。解の求めやすさは、問題を定式化する際にコスト関数と制約関数に割り当てる重みまたはパラメータに依存するため、適切なパラメータ調整が必要である。一般に、制約を満たすためには、コスト関数に割り当てられる重みは十分に小さくなければならない。一方、制約関数に割り当てられる重みが大きすぎると、低コストの解を得ることが難しくなる [8]。さらに、複数の制約が複数ある場合、その優先順位に従って重みを調整する必要がある。

QMF のハミルトニアンでは、 H_2 とその他の関数間にトレードオフの関係がある。 H_{cost} を基底状態にするためには $y_{\langle u,v \rangle} = 0$ が必要であるが、 H_2 を基底状態にするために $y_{\langle u,v \rangle} = 1$ にするように力が働く。さらに、 H_1 を基底状態にするためには、 $x_{n,s} = 1$ が必要になるが、 H_2 を基底状態にするために $x_{n,s} = 0$ にする力が働く。そのため、最適解を得るためには、これら 3 つの関数のバランスをとるような重みを設定する必要がある。

b. **ハードウェアパラメータ**

QA/QIA マシンは、それぞれ利用可能な量子ビットの数、量子ビットの接続グラフ、および設定可能なパラメータが異なる (2.2 節 表 2.1 参照)。そのため、それぞれのマシンで設定可能な様々なパラメータをチューニングする必要がある。例えば、全結合グラフではないマシンを利用して問題を解く場合には、2.2 節で述べた通り、頂点分割が必要になる。その際、分割した頂点間の相互作用 (図 2.3 の赤い辺) に、分割した頂点がそれぞれ同じ値になるように相互作用を設定する必要がある。この相互作用の値を小さく設定すると、分割された頂点それぞれが異なる値になる Chain break が発生する可能性が高くなる。一方、この相互作用を大きく設定すると、解くべきハミルトニアンの状態に影響し、最適解を得ることが難しくなる。そのため、Chain

break を引き起こさず、ハミルトニアンの状態に影響を与えない適切な相互作用を見つけた必要がある。この他にも、マシンによっては、アニーリングにかかる時間やアニーリングのステップ数が設定可能な場合もある。これらのパラメータを適切に設定することで、より最適解が得られやすくなる。

4.4.1 実験条件

前述 **a** のチューニングが、QIA マシンで得られる最適解の生成率にどの程度影響するかを調査した。実験では、Amplify AE を用いて網羅的に探索することで最適なパラメータを求める。Amplify AE の量子ビットは全結合であるため (2.2 節 表 2.1 参照), **b** で説明したような Chain break を考慮する必要はない。パラメータチューニングを行う QMF のハミルトニアンは次式である。

$$H = AH_{cost} + BH_1 + CH_2 \quad (4.28)$$

チューニングを容易にするため、式 (4.28) の各関数の最大係数が同じ大きさになるように関数 H' を展開した時の最大の係数値 $\max(H')$ で割ることで、各関数の大きさを正規化した。重みパラメータは $B = 1.00$ に固定し、 $A = 0.05 \sim 1.00$ は 0.05 ごと、 $C = 0.50 \sim 1.00$ は 0.10 ごとそれぞれの範囲で変化させた。表 4.2 に、実験に使用した行列のサイズとそれらの問題サイズに対する計算に必要な量子ビット数の平均値を示す。実験に用いる行列は非常に小さいが、式 (4.4) を KZFD-Ishikawa 法により次数削減することにより大量の量子ビットが必要になる。

表 4.2 パラメータの範囲と計算に必要な量子ビット数

matrix size	Required qubits
5 × 5	3904
6 × 6	11993
7 × 7	32268

今回の実験では、それぞれのパラメータセットに対して各 100 回計算し、次式で計算される最適解率と局所解率で評価した。

$$\text{最適解率} = \frac{\text{最適解の数}}{\text{計算回数}} \quad (4.29)$$

$$\text{局所解率} = \frac{\text{局所解の数}}{\text{計算回数}} \quad (4.30)$$

4.4.2 実験結果

この節では 2 種類の行列に対して実施した実験の結果を示す。

A. 消去順序を固定した行列

図 4.11 に示すような環状グラフに変換可能な行列を用いて得られた結果について述べる. この実験では, $N \times N$ の行列 (図 4.11(a)) に対して, 最初の $N - 3$ ステップの消去順序を固定し, 追加される辺がわかるようにする (図 4.11(b)). したがって, この実験での最適解は, 適切な頂点の消去順序が与えられ, 指定された辺が追加される解である.

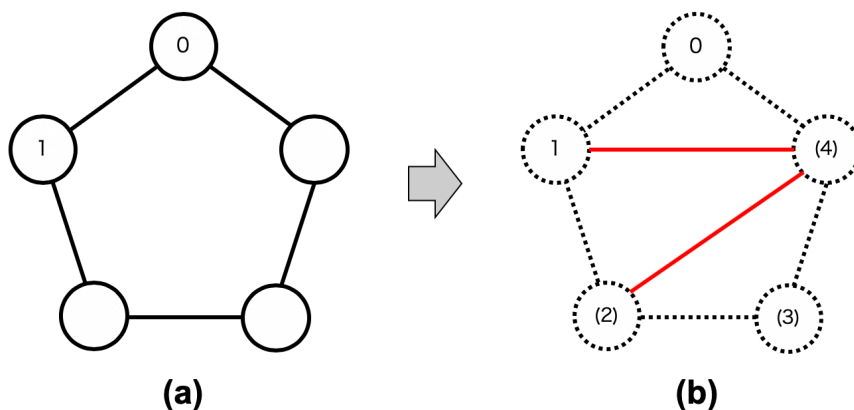


図 4.11 5×5 行列に対する消去順序を固定した消去グラフの例

図 4.12~ 図 4.14 に, 各問題サイズにおける最適解率のグラフを示す. 各グラフの折れ線は, 凡例に示した C の重みに対応している. これらの結果から, 行列サイズが大きくなるにつれて最適解率は低下し, 最適解率が高くなる A の値が大きくなる方向にシフトすることがわかる. また, C の値が大きくなるにつれて, 最適解率が高くなる A の値も大きい方にシフトすることがわかる. これらのことから, 行列のサイズが異なる問題で同じパラメータは使用できず, 最適解を得るためにはパラメータのチューニングが必要であることがわかる. さらに, この実験では, 行列サイズが 5×5 から 7×7 に増加するにつれて, 計算に必要な量子ビット数が約 8 倍に増加する. 量子ビット数の増加により, QIA マシンで最適解を見つけることができる確率を大幅に減少させることがわかる. また, これらの結果から, 最適解が得られるときのパラメータは, 5×5 行列と 6×6 行列では $A < C$ であるが, 7×7 行列では $A \geq C$ であることがわかる. 従って, 各関数に優先順位の高い順に重み付けをするだけでは不十分であると言える. これは, 前述 a で述べた一般論とは必ずしも一致しない結果である.

B. ランダムに生成した行列

対角要素が非零であり, それ以外の要素は $1/3$ の確率で非零であるランダムに生成された行列を用いた結果について述べる. QA や QIA はハミルトニアンエネルギーが最小に近い解を求めることができるが, 必ずしも最適解を求めるわけではない. そこで, 最適解だけでなく, 局所解もパラメータチューニングの評価の対象に入れる必

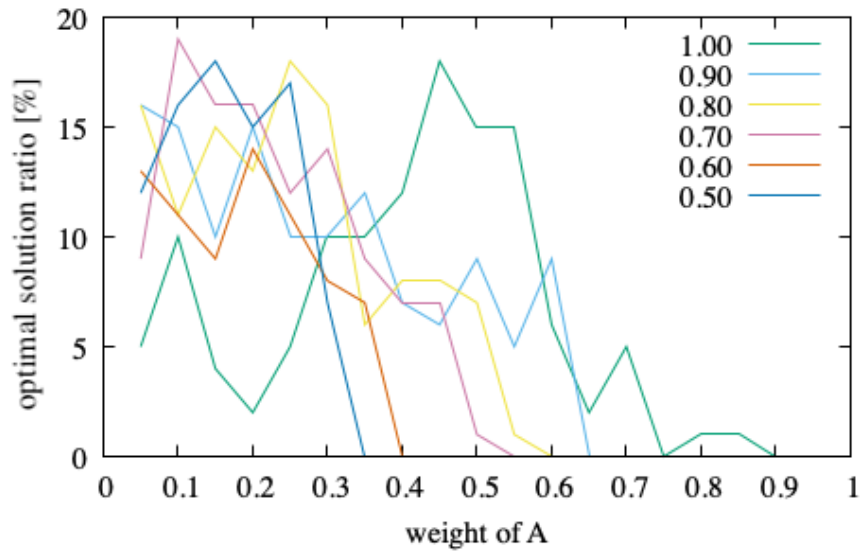


図 4.12 5 × 5 行列に対する最適解率

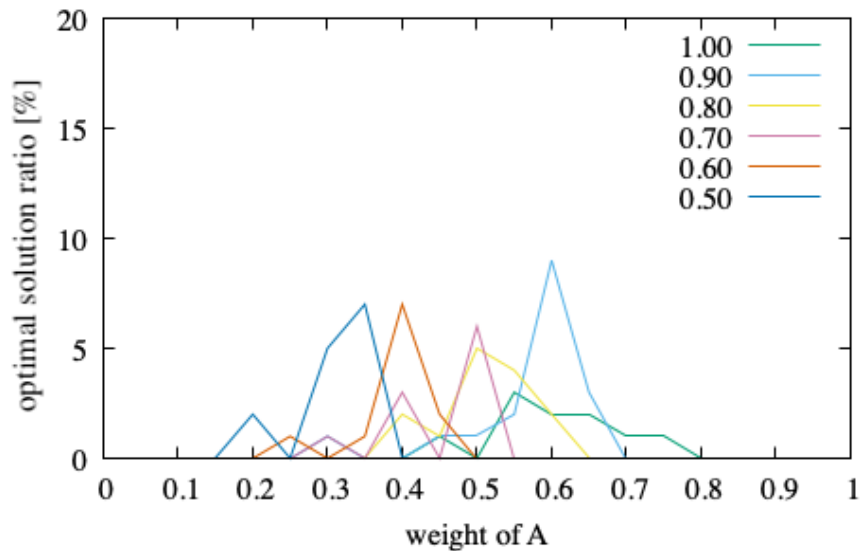


図 4.13 6 × 6 行列に対する最適解率

要がある．実験では，QMF を適用した場合（QMF）と，オーダリングなしの場合（NoOrder），従来手法の MD を適用した場合（MD）のフィルイン数を比較し，全てのオーダリング手法に対して同じ行列 100 個を利用した．図 4.15～図 4.17 に，各問題サイズにおける MD のフィルイン数以下の解率のグラフ，図 4.18～図 4.20 に各問題サイズにおける局所解（MD のフィルイン数以下の解も含む）率のグラフを示す．ここで，各グラフの折れ線は，凡例に示した C の重みに対応している．これらの結果から，5 × 5 行列の MD のフィルイン数以下の解は 94%，局所解は 14%，6 × 6 行列の MD のフィルイン数以下の解は 68%，局所解は 12%，7 × 7 行列の MD のフィルイン数以下の解は 27%，局所解は 8% と，パラメータチューニングによって解答率が変化することがわかる．これは，前述 A の実験結果と同様に，問題が大きくなるにつ

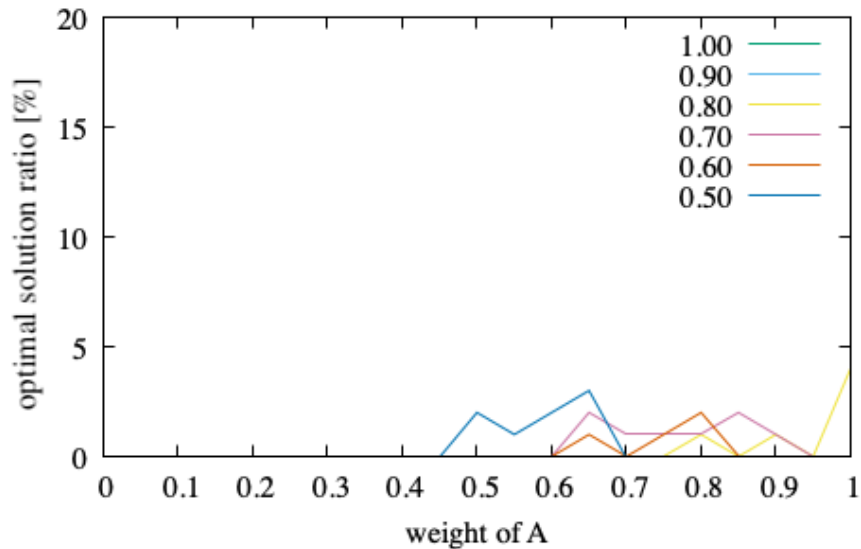


図 4.14 7×7 行列に対する最適解率

れて量子ビット数が増加するためである。また、この実験では、すべての MD のフィルイン数以下の解が MD と同じフィルイン数であった。これは問題サイズが MD で効果的に順序付けできる程度に小さいためである。さらに、これらの結果から、どのような行列サイズにおいても、MD のフィルイン数以下の解率は A の値が大きくなるにつれて増加し、 C の値が大きくなるにつれて減少することがわかる。ここから、最適なパラメータは $A = 0.85 \sim 1.00$, $B = 1.00$, $C = 0.5$ であると考えられる。前述 A の実験結果と同様に、満足させたい制約に高い重みを与えても最適な解が得られないことを示している。これは、前述の通り、QMF のハミルトニアンでは、 H_2 とその他の関数間にトレードオフの関係があることが理由にあると考えられる。制約とコスト関数の間だけでなく、制約間にもトレードオフの関係があり、実行可能解を得るためには制約間のパラメータの調整の方が大きな影響を与えられられる。

4.5 評価

この節では、4.2 節と 4.3 節の定式化方法による QUBO の変数の総数と解の質を評価する。

4.5.1 実験条件

この実験では、QMF の解を求めるために Fixstars の Amplify AE[14] を求解にかける時間である annealing time を 2000ms に設定して利用した。オーダリング対象の行列は、対角要素は非零であり、その他の要素は $1/3$ の確率で非零となるランダムに生成した各 200 個の $5 \times 5 \sim 10 \times 10$ 対称行列である。今回実験に用いた行列は、総当たりアルゴリズムにより最適解がわかっており、問題の 99% が従来手法の MD で最適解が得られる問題である。

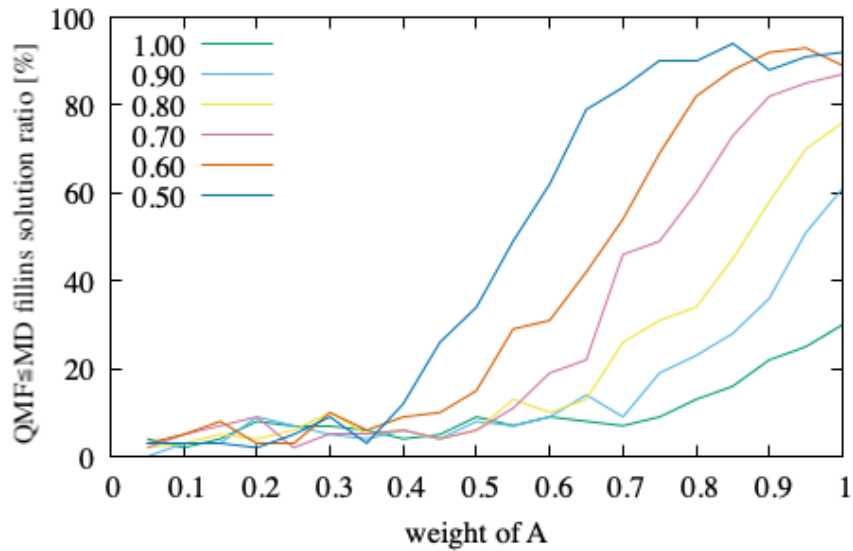


図 4.15 5 × 5 行列に対する MD のフィルイン数以下の解率

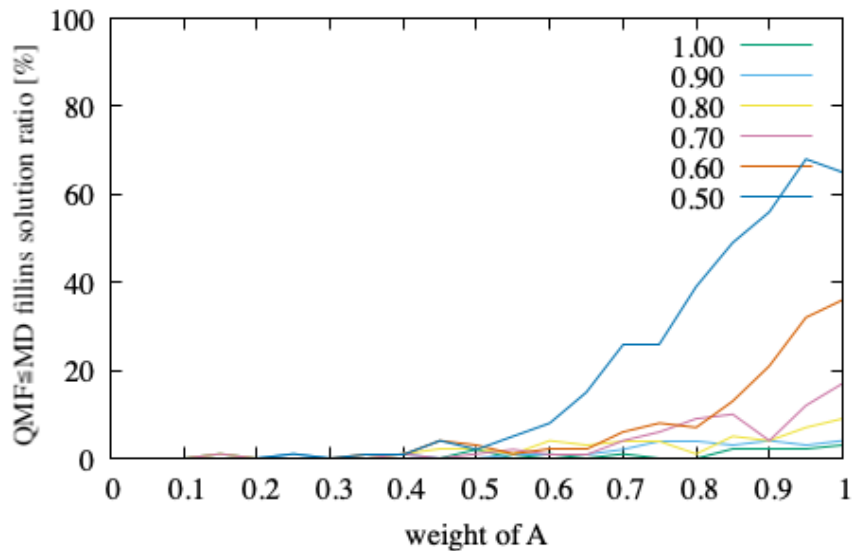


図 4.16 6 × 6 行列に対する MD のフィルイン数以下の解率

また、残りの 1% の問題も最適解のフィルイン数と MD によるオーダリングのフィルイン数が 1 しか違わないため、MD よりもフィルイン数が少ない解は最適解であることがわかる。

本実験に利用する QUBO のパラメータチューニングについて説明する。まず、次数削減を利用した QUBO は次式である。

$$H_a = H_{cost} + \alpha H_1 + \beta H_{2,a}^1 + \gamma H_{2,a}^2 \quad (4.31)$$

ここで α, β, γ は各関数に付加する重みである。関数 H' を展開した時の最大の係数値 $\max(H')$ を用いて、各関数の大きさを正規化した後、これらの重みの値を調整した。今回の定式化では、制約 2 の関数を制約 1 が満たされる前提で定式化したため、制約 1 に制約 2

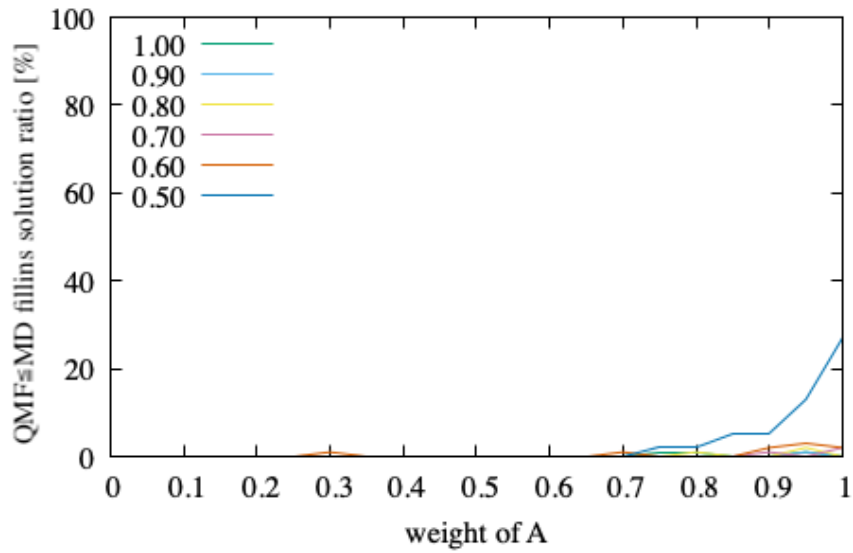


図 4.17 7×7 行列に対する MD のフィルイン数以下の解率

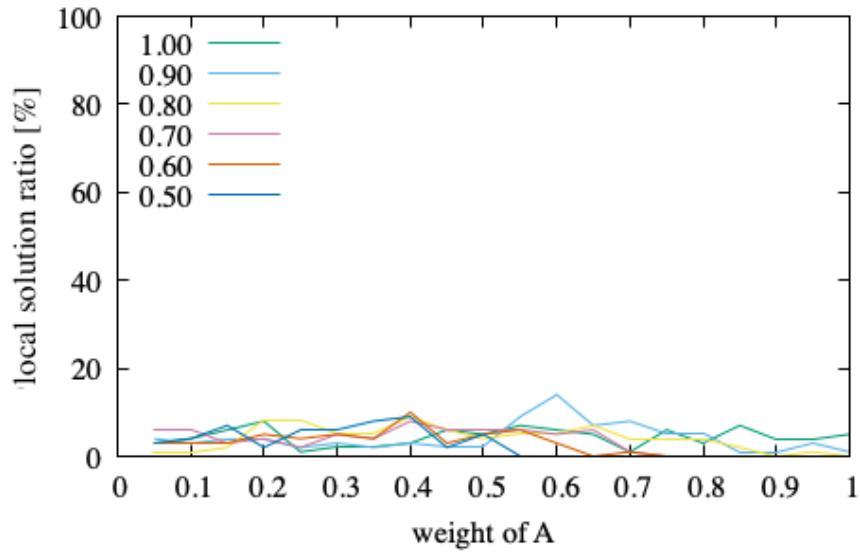


図 4.18 5×5 行列に対する局所解率

よりも大きな重みを付加した.

$$\alpha = \gamma \max(H_{2,a}^1) \max(H_{2,a}^2) \quad (4.32)$$

$$\beta = \gamma \max(H_{2,a}^2) \quad (4.33)$$

γ の値は 1 ~ 10 の範囲で 0.5 ごとに変化させ、フィルインが少ないオーダリングを多く得られた値を利用した. 次に、不等式を利用した QUBO は次式である.

$$H_b = H_{cost} + \sigma H_1 + \epsilon H_{2,b} \quad (4.34)$$

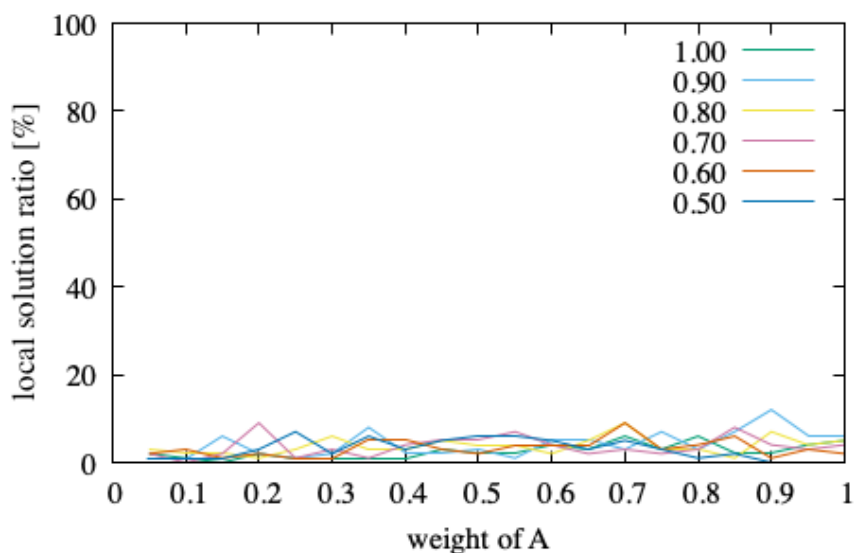


図 4.19 6×6 行列に対する局所解率

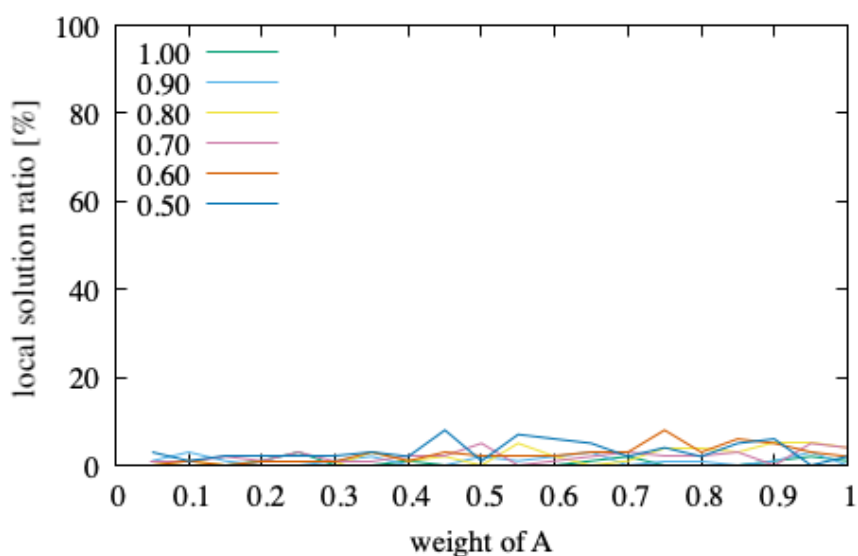


図 4.20 7×7 行列に対する局所解率

関数の重み σ も α, β と同様に設定した.

$$\sigma = \epsilon \max(H_{2,b}) \quad (4.35)$$

また, ϵ の値は 1 ~ 10 の範囲で 0.5 ごとに変化させ, フィルインが少ないオーダリングを多く得られた値を利用した. 制約が満たされているかどうかは, 次数削減を利用した QUBO の場合は制約関数 (4.2), (4.9), (4.10), 不等式を利用した QUBO の場合は制約関数 (4.2), (4.24) に得られた解 \boldsymbol{x} を代入し, エネルギー値が 0 になっていることを確認することで判定した.

4.5.2 最適解の得られやすさの評価

QMF, オーダリングなし, 従来手法の MD をそれぞれ同じ行列に適用し, 前進消去を実施したときのフィルイン数と計算時間を計測, 比較した. 表 4.3 に次数削減を利用した QUBO(4.31) の結果, 表 4.4 に不等式を利用した QUBO(4.34) の結果を示す. ここでは各 $N \times N$ 行列における, QMF で得られた制約を満たす解の数 (solutions), QMF とオーダリングなし, QMF と MD のフィルイン数の比較結果 (QMF-Non, QMF-MD), 各手法の平均フィルイン数 (avg. Fill-ins), 平均計算時間 (avg. execution time) を示している. 特に, QMF と各手法とのフィルイン数の比較結果は, QMF で得られた制約を満たす解の中からフィルイン数が $\text{QMF} < \text{Non}(\text{MD})$, $\text{QMF} = \text{Non}(\text{MD})$, $\text{QMF} > \text{Non}(\text{MD})$ となる解の数をそれぞれ求めた.

まず, 表 4.3 の次数削減を利用した QUBO(4.31) の結果から, QMF は行列サイズが大きくなるにつれて制約を満たす解が得られにくくなり, 8×8 行列までは低確率だが MD と同数以下のフィルインのオーダリングが得られることがわかる. また, 6×6 , 7×7 行列で MD では最適解が得られない問題でも最適解が得られている (太字の数字). しかし, 平均フィルイン数を比較すると, QMF はオーダリングなしの場合よりは少ないが, MD を適用した場合よりも多くなる. これは, QMF で最適解が得られる可能性が高くないことを示している. また, QMF の実行時間は, Amplify AE への通信時間による変化を除けば, 行列サイズに関わらず 3.86 秒程度で大きな変化がない. 一方, MD は行列サイズが大きくなると計算時間も長くなるが, 現状の QIA マシンで制約を満たす解が得られる行列サイズでは, QMF よりも MD の方がはるかに高速である.

表 4.3 次数削減による QMF の結果

N	α	solutions	QMF - NoOrder			QMF - MD			avg. Fill-ins			avg. execution time [s]		
			<	=	>	<	=	>	QMF	MD	NoOrder	QMF	annealing time	MD [$\times 10^{-5}$]
5	2.0	200	86	114	0	0	200	0	0.1	0.1	0.8	3.86	1.91	1.97
6	1.4	168	97	62	9	2	136	30	0.4	0.3	1.7	5.03	1.90	2.48
7	0.8	50	31	14	5	1	30	19	1.0	0.5	2.7	4.86	1.88	3.14
8	1.6	7	6	0	1	0	3	4	2.3	1.1	6.6	3.81	1.85	3.00
9	1.6	2	1	1	0	0	0	2	5.5	2.5	7.5	3.89	1.96	5.02
10	-	0	-	-	-	-	-	-	-	-	-	-	-	-

次に, 表 4.4 の不等式を利用した QUBO(4.34) の結果から, 次数削減を利用した QUBO(4.31) よりも, 大きな行列で制約を満たす解が得られやすいことがわかる. さらに, 6×6 , 7×7 , 10×10 行列では MD で最適解が得られない問題でも最適解が得られている. ただし, オーダリングなしや MD よりもフィルイン数が多くなってしまう場合も少なくはない. そのため, 平均フィルイン数を比較すると, QMF はオーダリングなしの場合よりは少なく, MD を適用した場合よりも多くなる. また, QMF の実行時間 Amplify AE への通信時間による変化を除けば, 行列サイズに関わらず 3.80 秒程度で大きな変化がない.

表 4.4 不等式制約による QUBO の結果

N	α	solutions	QMF - NoOrder			QMF - MD			avg. Fill-ins			avg. execution time [s]		
			<	=	>	<	=	>	QMF	MD	NoOrder	QMF	annealing time	MD [$\times 10^{-5}$]
5	2.0	200	86	114	0	0	200	0	0.1	0.1	0.8	3.87	1.91	1.97
6	1.8	190	117	70	3	2	164	24	0.4	0.3	1.7	5.77	1.90	2.51
7	1.4	130	84	33	13	1	76	53	1.1	0.5	2.7	4.53	1.85	3.17
8	2.0	83	53	13	17	0	21	62	2.7	1.0	4.6	3.79	1.86	3.86
9	1.2	22	11	3	8	0	3	19	6.3	1.3	8.1	3.73	1.83	4.76
10	1.8	10	5	0	5	1	0	9	8.2	2.7	9.0	3.78	1.85	5.83

以上の結果から、次数削減を利用した QUBO(4.31) と不等式を利用した QUBO(4.34) のどちらの定式化手法でも QMF で最適解のオーダリングが得られ、QMF による疎行列のオーダリングが実現できたことがわかる。ただし、どちらの定式化手法でも行列サイズが大きくなるにつれ、制約を満たす解や最適解が得られにくくなった。これは、問題サイズに対する変数の数がまだまだ多く、現状の QIA マシンでは上手く解が得られないことが原因であると考えられる。また、小さい行列において効果的にオーダリングが可能な MD で得られなかった最適解を QMF で得た。このような結果となる行列は、あるステップにおいて最小次数となる頂点が複数ある行列であった。これらの行列に対して、MD では頂点選択の基準がないのに対して、QMF では全ステップを通して最小フィルインとなるオーダリングを求めることができる。さらに、計算時間に関して、MD は行列サイズが大きくなるにつれて計算時間も増加した。一方、QMF は行列サイズが大きくなってもほぼ同じ計算時間で解を得ることができたが、問題サイズが大きくなると制約を満たす解が減少している。一般に、QA の求解速度は問題サイズにほとんど依存しないが、問題サイズが大きくなると制約を満たす解が見つかりにくくなる。今回の実験では QIA マシンを利用したため、問題サイズが大きくなると計算に必要な時間も大きくなる可能性があるが、将来的に QA を利用することで、問題サイズが変化しても一定の計算時間で QMF の計算が可能になると考えている。しかし、今回実験した行列サイズでは MD の方が大幅に高速であった。QMF の計算時間は、アニーリング時間と、QUBO の入力形式の変換や出力から解への変換などの前後処理で半分ずつ時間を費やしていた。特に、QUBO の入力形式の変換する処理は $O(N^2)$ である [52]。QA マシンを利用した多くの応用研究に見られるように、問題サイズが大きくなりアニーリング時間よりも QA/QIA の入力形式への変換などの時間の増加率が大きく、QMF の計算時間におけるボトルネックとなることが予想される。今後、前後処理の並列化などによる QUBO 生成の高速化により、QMF でより高速に解が得られることが期待される。

4.5.3 QUBO の変数接続数

まず、前述の評価で利用した行列のうち 100 個の 5×5 行列に対する各 QUBO の変数の接続数を表 4.5 に示す。これらの表から、変数の接続数は次数削減を利用した QUBO(4.31) の方が不等式を利用した QUBO(4.34) よりも少ないことがわかる。一般的に変数の接続数が少ない方が、疎結合グラフへの埋め込みの際の補助変数の数が少なくなり、QA で解が得

られやすいとされている。

表 4.5 変数の接続数

定式化手法	Min	Max	Avg.	Median
HOBO+ 次数削減	5	148	12	8
不等式 + スラック変数	9	315	24	15

この評価では全結合の物理グラフを持つ Amplify AE で評価を行ったため、不等式を利用した QUBO(4.34) の方が次数削減を利用した QUBO(4.31) よりも最適解を多く得たが、疎結合グラフを物理グラフとしてもつ QA/QIA マシンでは次数削減を利用した QUBO(4.31) の方が最適解が得られやすくなることがある。例えば、Djidjev[46] の研究では、不等式で定式化される問題を HOBO で定式化し、疎結合グラフで量子ビットが接続されている D-Wave マシンで解いた。この実験では、不等式をスラック変数を導入して QUBO に定式化した場合よりも、HOBO を次数削減して定式化した QUBO の方が埋め込みがしやすく、質の良い解が得られることを示した。ただし、この論文では、不等式から HOBO へ変換する一般的な手法は示されていない。これに対して、4.3 節の HOBO から不等式への変換を逆向きに適用して定式化することが有効である。現状の QA マシンを利用する場合は、不等式を利用するよりも高次式を利用して定式化した方が良いだろう。しかし、今後の技術の発展により QA マシンは量子ビットを全結合で接続されることが予想され、本論文で述べたような不等式を用いた定式化の方が有効になると考えられる。また、付録には、本章と異なる定式化手法による QMF の評価と、QUBO 行列の解析結果を示している。

4.5.4 QMF に対する線形ソルバの利用

4.3 節では、QMF を不等式を利用して QUBO に定式化した。不等式を利用して QMF を定式化する場合、次のように 1 次の線形問題として定式化することもできる。

$$\text{minimize} \quad \sum_{\langle u,v \rangle \in \bar{E}} y_{\langle u,v \rangle} \quad (4.36)$$

$$\text{subject to} \quad \sum_{s=0}^{N-1} x_{n,s} = 1 \quad \forall n = 0, 1, \dots, N-1 \quad (4.37)$$

$$\sum_{n=0}^{N-1} x_{n,s} = 1 \quad \forall s = 0, 1, \dots, N-1 \quad (4.38)$$

$$\begin{aligned} (1 - y_{\langle u, v \rangle}) + x_{n,s} + \left(1 - \sum_{k=0}^s x_{u,k}\right) + \left(1 - \sum_{k=0}^s x_{v,k}\right) \\ + y_{\langle n, u \rangle} + y_{\langle n, v \rangle} \leq 5 \\ \forall \langle u, v \rangle \in \bar{E}, \quad \forall n \notin \{u, v\}, \quad \forall s = 0, 1, \dots, N-1 \end{aligned} \quad (4.39)$$

問題を 1 次式で定式化できる場合、従来利用されてきた線形最適化ソルバにより解を求めることができる。本節では線形ソルバ CPLEX[53] を利用した QMF によりオーダリングを求めた結果を示す。計算には Intel Core i7, 4 コア, 8 スレッドの CPU を搭載したマシンを利用し、CPLEX は 8 スレッドの並列計算により解を求めた。また、評価に利用した行列は 4.5.1 節に示した行列のうち 100 個である。

まず、図 4.21 に線形ソルバによる QMF と MD のフィルイン数の比較を示す。ここから、問題サイズが大きくなるにつれて、QMF の方が MD よりもフィルイン数が少ない解が増加することがわかる。問題サイズが大きくなるにつれて、MD で最適解が得られなくなる行列が増加するのに対し、そのような行列に対しても QMF で最小フィルインの解が得られていると考える。

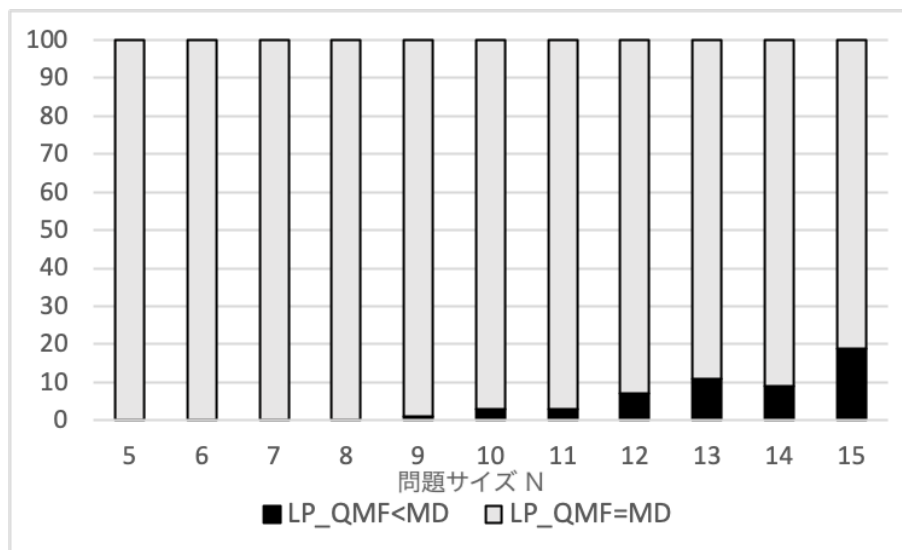


図 4.21 フィルイン数の比較

次に、図 4.22 に線形ソルバによる QMF と QA による QMF, MD の平均計算時間を示す。ただし、QA による QMF の計算時間は表 4.4 のものであるため、実行環境と計算回数異なることに注意が必要である。また、この図には、3 章で述べた最小フィルインの厳密解法の計算量である $O(1.8899^N)$ の傾きを黒点線で示している。まず、線形ソルバによる QMF では問題サイズが大きくなるにつれて、計算時間も大きくなることがわかる。線形ソルバによる QMF のグラフの傾きと最小フィルインの厳密解法の計算量 $O(1.8899^N)$ の点線と比較すると、CPLEX による QMF の方が傾きが大きい、つまり、計算時間の増加量が大きいことがわかる。これは、CPLEX による QMF の計算時間には、QUBO 生成等の問題設定の時間も含まれていることが原因であると考えられる。さらに、線形ソルバによる QMF よりも MD の方が 100 倍以上高速になることがわかる。これは、QMF は最小フィルインのオーダリングを厳密に求めるのに対し、MD はヒューリスティックに求めるためである。また、QA による QMF と線形ソルバによる QMF を比較すると、問題サイズが現状の QA/QIA マシンで実行可能解が得られる $N = 10$ までであれば QA よりも線形ソルバの方が高速に解を得るが、それ以上の大きさの問題では QA の方が高速に解を得る可能性が高い。今後

QA/QIA マシンでより大きな問題に対しても高精度の解が得られるようになったとき，従来の最適化ソルバよりも QA/QIA マシンの方が QMF に適した解法になると考える。

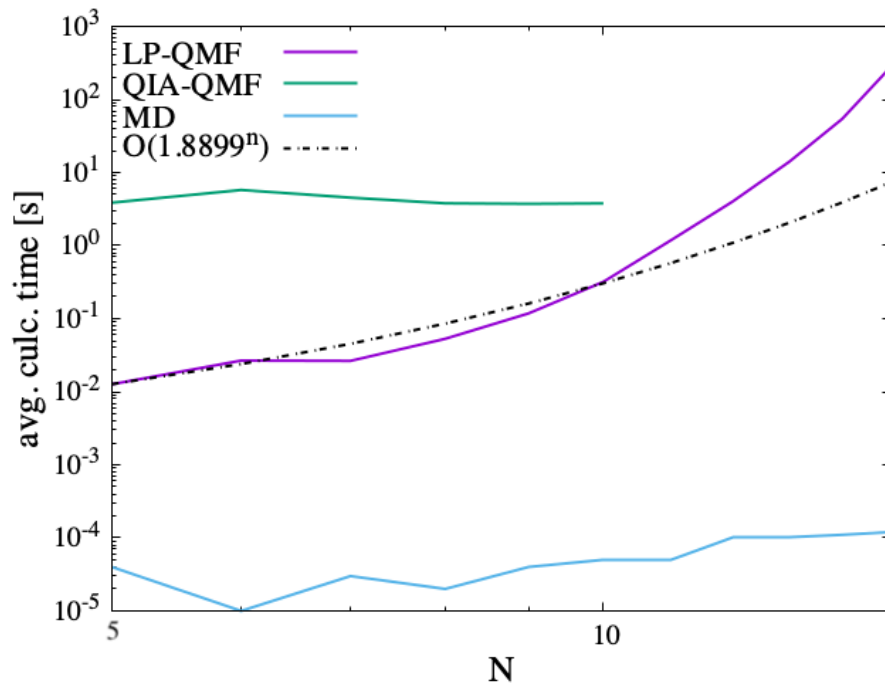


図 4.22 計算時間の比較

第 5 章

QMF の問題緩和とハイブリッドオーダリング

4 章では、QMF を QUBO に定式化する際の補助変数を削減し、動作実験の結果から QMF で従来手法で最適解が得られない問題でも最適解が得られる可能性を示した。しかし、現在の QA/QIA マシンの限られた量子ビット数では、 10×10 以下の小さなサイズの行列問題にしか QMF を適用できなかった。このような小さな行列の問題は元々フィルインの発生が少なく、従来手法でも高速にフィルイン最小のオーダリングを求めることができる。そこで、本章では問題緩和とハイブリッドオーダリングの 2 つの観点からそれぞれ大きな行列の問題を扱う方法を検討する。

5.1 問題緩和

本節では QMF を緩和することにより、大きな行列の問題を扱う方法を提案する。本節の内容は関連論文 [b] を基にしている。

5.1.1 概要

疎行列に対する直接解法の前進消去では、最初の数ステップで多くのフィルインが発生する傾向がある。図 5.1 に QMF を 10 個の $N \times N$ 対称行列 ($1/3$ の確率で非零要素) に適用した場合における最初と最後の半分のステップでそれぞれ発生する平均フィルイン数を示す。ここから、フィルインは前半のステップで大半が発生することがわかる。それにより、最初の数ステップの順序を見つけるだけでもフィルイン数の削減が期待できる。

最初の数ステップだけオーダリングすることにより、QMF の計算に必要な変数の総数の大幅な減少が期待できる。そのため、現状の QA/QIA マシンで解くことができる行列サイズを大きくすることができる。QMF の最初の数ステップのオーダリングのみを求めるため

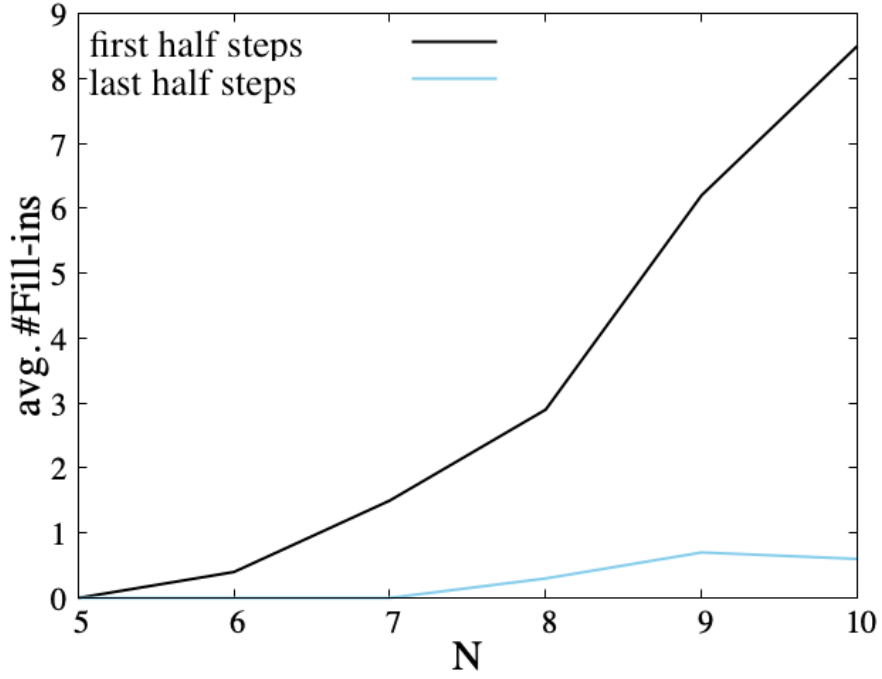


図 5.1 最初と最後の半分のステップでそれぞれ発生する平均フィルイン数 [b]

に，制約 1 の QUBO を次のように変更する．

$$H'_1 = \sum_{n=0}^{N-1} \left\{ \sum_{s=0}^{S-1} x_{n,s} \left(\sum_{s=0}^{S-1} x_{n,s} - 1 \right) \right\} + \sum_{s=0}^{N-1} \left(\sum_{n=0}^{N-1} x_{n,s} - 1 \right)^2 \quad (5.1)$$

ここで N は問題サイズ， S はオーダリングするステップ数である．最初の数ステップだけをオーダリングする場合，図 5.2 のように，消去される頂点と消去されない頂点が存在する．したがって，各頂点 n において消去ステップを表す変数 $x_{n,s}$ の和が 0 か 1 のとき最小値 0 をとる関数として定式化できる．今回は，最初の数ステップで消去される行/列の要素のみを入れ替え，その他の行/列は入れ替えない．行/列が入れ替えられることなく消去されることは，消去グラフの頂点番号の昇順で消去されることと等価である．

QIA マシンで基底状態を求める QUBO は，4 章の次数削減と不等式による定式化のそれぞれを利用して定式化される．

$$H'_a = H_{cost} + \alpha H'_1 + \beta H_1^{2a} + \gamma H_2^{2a} \quad (5.2)$$

$$H'_b = H_{cost} + \sigma H'_1 + \epsilon H_{2b} \quad (5.3)$$

ここで，各関数につける重みは，4 章のパラメータ設定 (4.32)，(4.35) と同様に設定する．ただし， γ と ϵ の値は 1 ～ 10 の範囲を 0.5 ごとに変化させ，最もフィルイン数が少ないオーダリングが得られた値を評価に利用した．また，現状の QA/QIA マシンで解くことができる問題サイズでは，5 ステップ以上の QMF では変数削減数は期待できず，後半のステップ

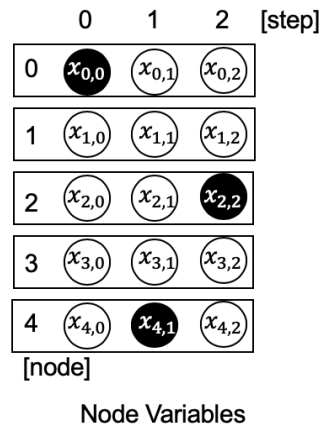


図 5.2 ハイブリッドオーダリングの QMF における変数設定例 [b]

で発生するフィルインの数は少ない。そのため、評価では最初の 3 ステップのオーダリングを求めることとした。

5.1.2 評価結果と考察

問題緩和により現状の QIA マシンを利用した QMF で解くことができる行列の大きさと得られる解の質を評価した。この評価も非ゼロ要素が 1/3 の確率で出現するランダム対称行列 100 個を利用し、Fixstars の Amplify Annealing Engine[14] をアニーリング時間は 2000ms に設定して解を求めた。

A. 計算に必要な変数の総数

最初の 3 ステップと全ステップの平均量子ビット数の比較を図 5.3 に示す。この図には、i) 次数削減を利用した QUBO(5.2) の 3 ステップのみの QMF, ii) 次数削減を利用した QUBO(5.2) の全ステップの QMF, iii) 不等式制約を利用した QUBO(5.3) の 3 ステップのみの QMF, iv) 不等式制約を利用した QUBO(5.3) の全ステップの QMF を示している。ただし、次数削減を利用した QUBO(5.2) と不等式制約を利用した QUBO(5.3) のどちらの定式化を用いても、必要となる変数の数に大きな違いはないため、i と iii, ii と iv, 全ステップ QMF に対する 3 ステップのみの QMF の平均量子ビット数である ratio:i/ii と ratio:iii/iv が重なっていることに注意が必要である。次数削減を利用した QUBO(5.2) と不等式制約を利用した QUBO(5.3) のどちらの QMF でも、問題緩和により量子ビット数が大幅に減少した。特に、問題サイズ N が大きいほど、量子ビット数の減少は大きくなり、 $N = 15$ では、緩和前の問題の約 20% 程度の量子ビットで計算が可能である。この結果から、問題緩和により変数の数を削減することは、量子ビット数に限りがある QA/QIA マシンを利用してより大きなサイズの問題を解くために有効な手段であると考えられる。

B. フィルイン数

同じ 100 個の対称疎行列に対して、オーダリングなし、MD、全ステップの QMF、

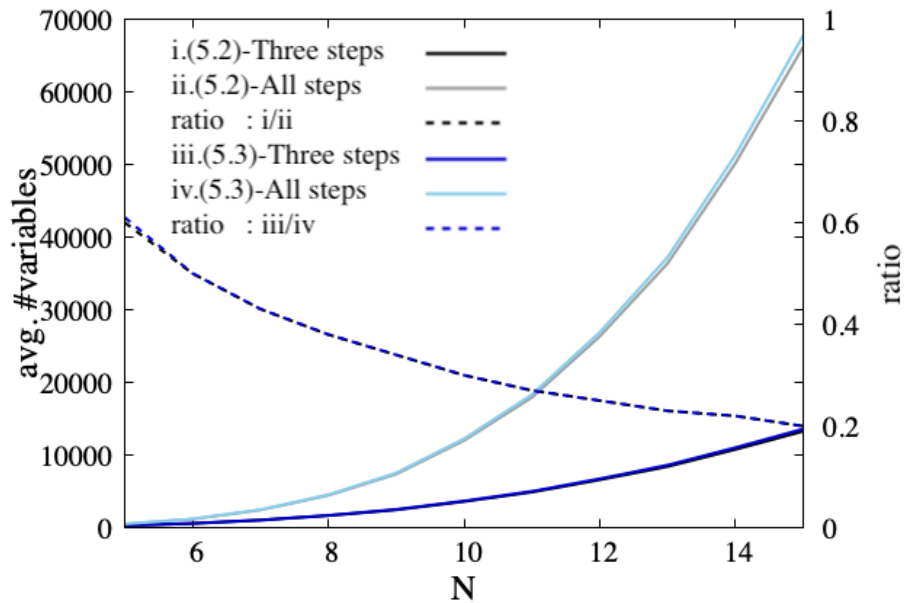


図 5.3 計算に必要な変数の総数の平均数 [b]

数ステップのみの QMF をそれぞれ適用した後，前進消去を実施した場合に発生するフィルイン数を比較する．QMF は次数削減を利用した QUBO(5.2) と不等式制約を利用した QUBO(5.3) の定式化も比較した．評価は，

- 1) QMF で得られた制約を満たす解の数
- 2) 各手法のフィルイン数の平均値 (avg.Fill-ins)
- 3) QMF と各手法の間のフィルイン数の比較結果 (QMF とオーダリングなし (NoOrder), QMF と MD をそれぞれ比較)

の 3 つの観点から行なった．QMF と各手法のフィルイン数の比較は，制約を満たす解のうち，フィルイン数が $\text{QMF} < \text{NoOrder (MD)}$ ， $\text{QMF} = \text{NoOrder (MD)}$ ， $\text{QMF} > \text{NoOrder (MD)}$ の数をそれぞれ示す．例えば，表 5.1 の「QA-NoOrder Fill-ins」の「Three steps」の欄から， $N = 5$ の場合，QMF はオーダリングなしで得られた解よりもフィルイン数が少ない解を 36 個，フィルイン数が同じになる解を 64 個得たことがわかる．ただし，この評価に利用した行列のうち 98~99 個の行列は MD で最適解が得られることを，全探索アルゴリズムにより確認している．

まず，1) の結果を図 5.4 に示す．この結果から，どの QMF でも行列サイズが大きくなるにつれて制約条件を満たす解を得ることが難しくなることがわかる．次数削減を利用した QUBO(5.2) では，最初の 3 ステップのみオーダリングした場合は 10×10 行列まで，全ステップをオーダリングした場合は 8×8 行列まで制約を満たす解が得られた．さらに，不等式を利用した QUBO(5.3) では，最初の 3 ステップのみオーダリングする場合は 15×15 行列まで，全ステップをオーダリングした場合は 10×10 行列まで制約を満たす解が得られた．以上の結果から，問題緩和により，実際に解かれる問題サイズが小さくなり，より大きな行列で制約を満たす解が得られる

ことがわかる．さらに，次数削減を利用した QUBO(5.2) よりも 不等式を利用した QUBO(5.3) の方が制約を満たす解を得やすかった．ただし， 16×16 以上の行列では Amplify AE で計算可能な変数の総数となる問題サイズであっても，どちらの定式化も制約を満たす解が得られなかった．

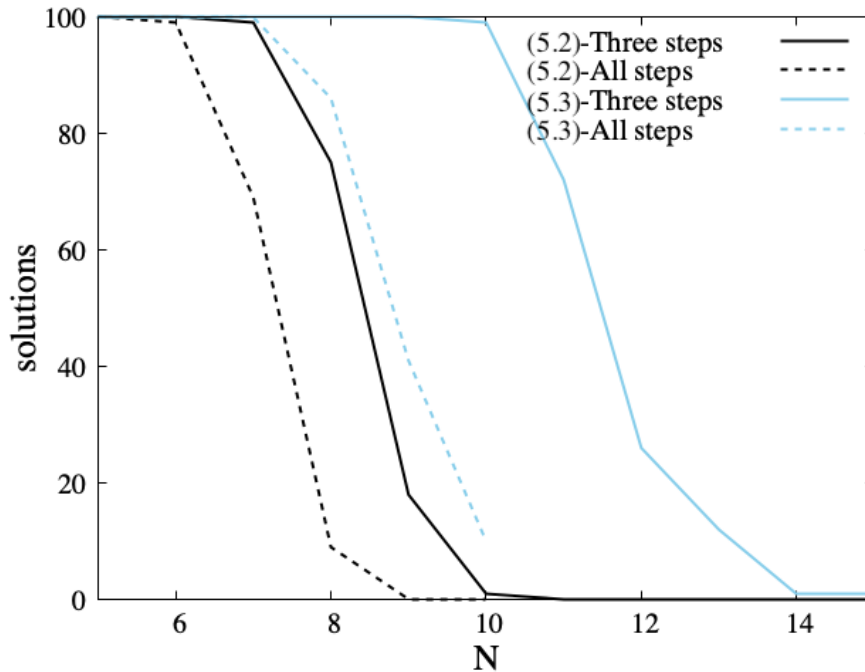


図 5.4 実行可能解の総数 [b]

次に，次数削減を利用した QUBO(5.2) を用いた QMF の 2) と 3) の結果をそれぞれ図 5.5 と表 5.1 に示す．図 5.5 から QMF の平均フィルイン数は，最初の 3 ステップのみと全てステップどちらの場合もほとんど同じである（この図では “All steps-QMF” と “Three steps-QMF” が重なっているのに注意）．また，QMF の平均フィルイン数は NoOrder より小さくなるが，MD より大きくなるのがわかる．また，表 5.1 から全ステップの場合よりも最初の 3 ステップのみの QMF の方が，MD と同数のフィルインの解が多く得られる傾向があることがわかる．これらの結果から，次数削減を利用した QUBO(5.2) を用いた QMF では，問題緩和しない場合と同等以上の質の解がより多く得られることがわかる．ただし，この実験では，MD よりも少ないフィルイン数の解を得ることはできなかった．

さらに，不等式を利用した QUBO(5.3) を用いた QMF の 2) と 3) の結果をそれぞれ図 5.6 と表 5.2 に示す．図 5.6 から QMF は，最初の 3 ステップのみの方が全てステップオーダリングした場合よりも平均フィルイン数が少なくなることがわかる．また，その他の手法を適用した場合の平均フィルイン数と比較すると，全てステップオーダリングした場合は問題サイズが大きくなるにつれて NoOrder の値に近づいていくが，最初の 3 ステップのみをオーダリングした場合は問題サイズが大きくなっても NoOrder よりも小さく MD よりも大きい値になることがわかる．さらに，表 5.2

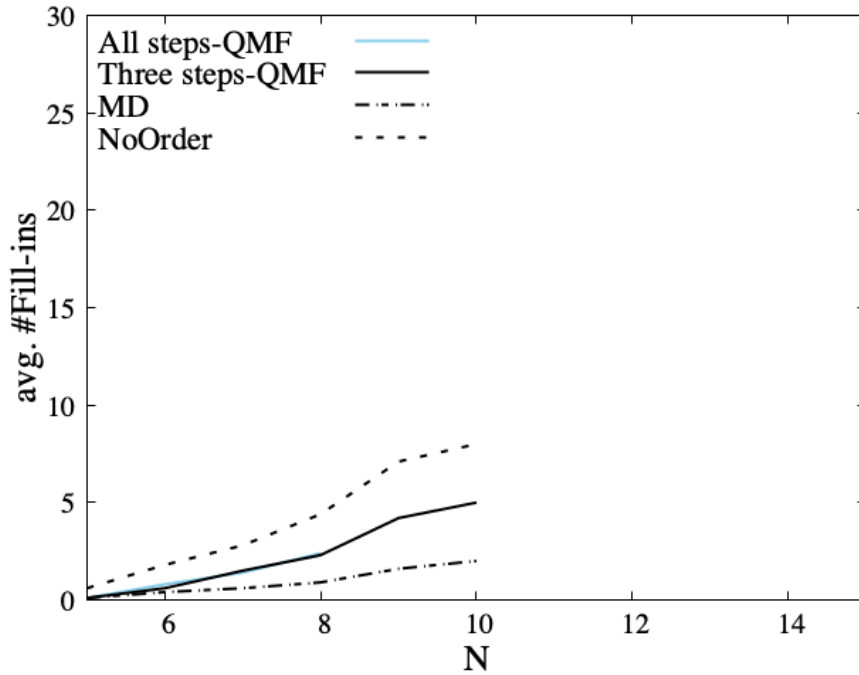


図 5.5 次数削減を利用した QUBO(5.2) を用いた QMF と他手法の平均フィルイン数 [b]

表 5.1 次数削減を利用した QUBO(5.2) を用いた QMF と他手法のフィルイン数の比較 [b]

N	QMF-NoOrder Fill-ins			QMF-MD Fill-ins								
	Three steps			All steps								
	<	=	>	<	=	>						
5	36	64	0	35	65	0	0	100	0	0	99	1
6	54	40	6	47	37	15	0	77	23	0	63	36
7	50	42	7	44	17	8	0	44	55	0	35	34
8	53	19	3	7	1	1	0	26	49	0	3	6
9	16	2	0				0	2	16			
10	1	0	0				0	0	1			

から、全ステップの場合よりも最初の 3 ステップのみの QMF の方が制約を満たす解と NoOrder よりもフィルイン数が少ない解が多くなることがわかる。しかし、問題サイズが大きくなると制約を満たす解は得られるものの、MD よりもフィルイン数が大きくなる解が多く得られる傾向があり、MD よりも少ないフィルイン数の解を得ることはできていない。以上の結果から、不等式を利用した QUBO(5.3) を用いた QMF において、制約を満たし、オーダリングなしよりもフィルイン数が少なくなるオーダリングが多く得られる点で問題緩和は有効であると考えられる。

以上の結果から、提案手法による問題緩和が QMF に対して有効であり、不等式を利用した QUBO(5.3) の方が、次数削減を利用した QUBO(5.2) よりも問題緩和の効果が大きいことが分かった。しかし、QMF 適用した際のフィルインの平均数は、オー

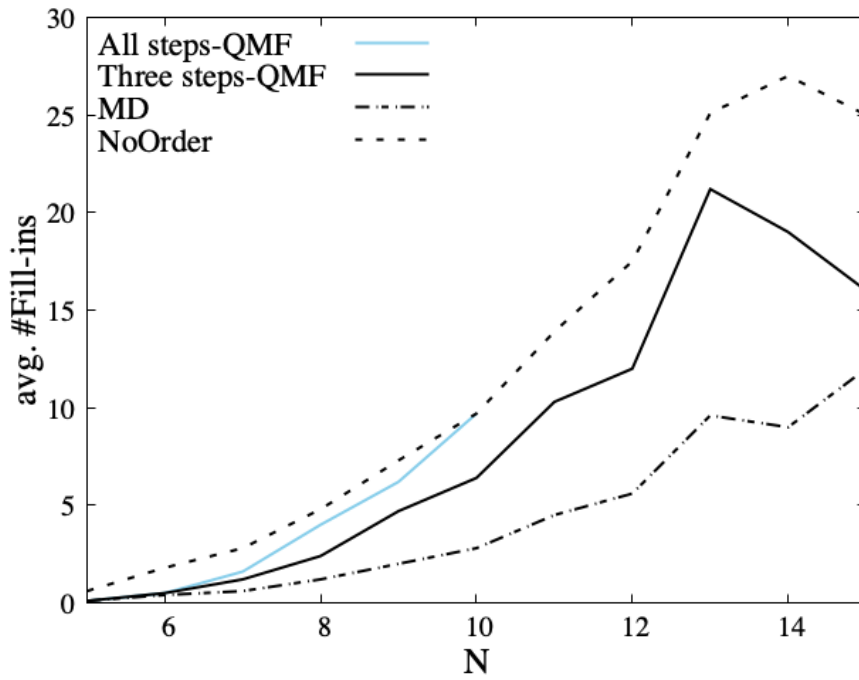


図 5.6 不等式を利用した QUBO(5.3) を用いた QMF と他手法の平均フィルイン数 [b]

表 5.2 不等式を利用した QUBO(5.3) を用いた QMF と他手法のフィルイン数の比較 [b]

N	QMF-NoOrder Fill-ins			QMF-MD Fill-ins		
	Three steps	All steps		Three steps	All steps	
	< = >	< = >		< = >	< = >	
5	36 64 0	36 64 0		0 100 0	0 100 0	
6	56 44 0	54 42 4		0 90 10	0 86 14	
7	64 33 3	56 21 23		0 60 40	0 41 59	
8	74 22 4	42 13 31		0 35 65	0 11 75	
9	76 17 7	27 1 13		0 11 89	0 3 38	
10	76 13 10	4 0 6		0 7 92	0 0 10	
11	60 8 4			0 1 71		
12	21 2 3			0 0 26		
13	10 0 2			0 0 12		
14	1 0 0			0 0 1		
15	1 0 0			0 0 1		

ダリングを適用しない場合よりも小さく，MD を適用した場合よりも大きい．これは，QA が常に最適解を与えるとは限らないため，サイズが小さい場合に有効な MD 法よりも良いオーダリングができないことによる．今後，QA/QIA マシンでより大きな行列が解けるようになれば，QMF で MD よりも効果的なオーダリングができるようになる可能性がある．より大きなサイズの行列を計算可能になったとき，数ステップしかオーダリングしないことによるフィルイン数の変化についてより詳細な議論が必要である．

5.2 QMF と MD のハイブリッドオーダリング

5.1 節では最初の数ステップ分のみをオーダリングすることにより問題緩和をした．これにより，緩和する前よりも若干大きな問題サイズでも QA/QIA で解を求めることを可能にしたが，計算時間は従来手法の MD の方が明らかに高速になった．一方，MD は最小次数が複数ある場合，辞書順やランダムなど任意に選択されており，明確な選択指針がない．そこで，本節では MD と QMF のハイブリッドオーダリングを検証する．このオーダリング手法は，基本は MD でオーダリングを行うが，最小次数頂点が複数ある場合に数ステップのみの QMF でオーダリングを行う手法である．ただし，QMF で 1 ステップずつオーダリングする場合と MD は等価ではないことに注意が必要である．これは，QMF は追加される辺の本数が最小になる頂点を選択するのに対し，MD では最小次数の頂点を選択するためである．

古典コンピュータによる総当たりアルゴリズムにより追加される辺の本数を数ステップ先まで確認して頂点を選択する場合の計算時間を図 5.7 に示す．この図 5.7 から，古典コンピュータによる総当たりアルゴリズムで計算する場合，ステップ数や問題サイズが増加すると，計算時間の増加率も大きくなるが，QA で計算する場合は問題サイズによって計算時間が変化しないことがわかる．さらに，3 ステップで 22×22 ，5 ステップで 11×11 以上の行列で QA で計算した方が高速になることがわかる．

ハイブリッドオーダリングの QMF では，図 5.8 に示すようにオーダリング対象の頂点を 2 通り設定した．図 5.8(a) は最初の 1 ステップ目は最小次数頂点から消去する頂点を選択し，それ以降のステップでは全ての頂点から消去する頂点を選択するような変数設定である．図 5.8(b) は数ステップ全てにおいてある時点での最小次数頂点から消去する頂点を選択する変数設定である．また，利用するオーダリング結果は，1. 全ステップ，2. 1step のみの 2 通りである．

評価は，対角要素が零ではなく，それ以外の要素は $1/3$ の確率で非零要素となる $N \times N$ 対称行列 100 個に対して，ハイブリッドオーダリング (HYB)，QMF，MD をそれぞれ適用したときとオーダリングなし (NoOrder) のときのフィルイン数を計測した．QMF の計算には，Amplify AE を利用し，アニーリングにかかる時間である annealing time はハイブリッドオーダリングの QMF は 1ms，全ステップ QMF は 2000ms に設定して利用した．

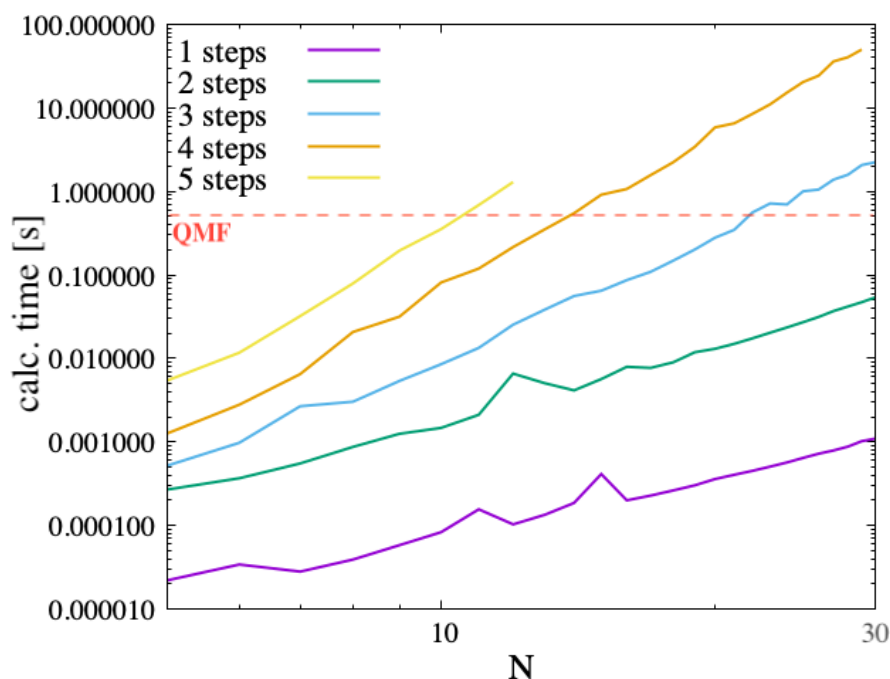


図 5.7 総当たりアルゴリズムの計算時間

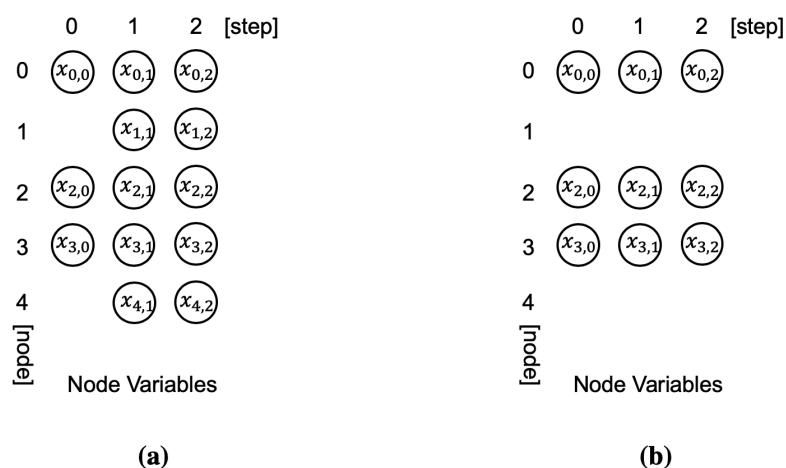


図 5.8 ハイブリッドオーダリングにおける QMF のオーダリング対象頂点

図 5.9 に各オーダリング手法における平均フィルイン数のグラフを示す。ただし、a-1, a-2, b-2 はほぼ同じ平均フィルイン数となったため、グラフの折線が重なっていることに注意してほしい。この図 5.9 から、b-1 の 8×8 行列と 9×9 行列以外で HYB により MD 以下の平均フィルイン数となることがわかる。この結果から、HYB により従来手法よりも最小フィルインに近い解が得られると考える。また、QMF は最小フィルインとなる解を得るため、HYB よりも QMF の方がフィルイン数が少ない解を得る可能性が高いと予想していたが、そのような解は僅かであった。これは、QMF で最適解が得られにくいことが原因である。

次に、平均計算時間を図 5.10 に示す。QMF の 20% の時間で計算が可能であるが、MD

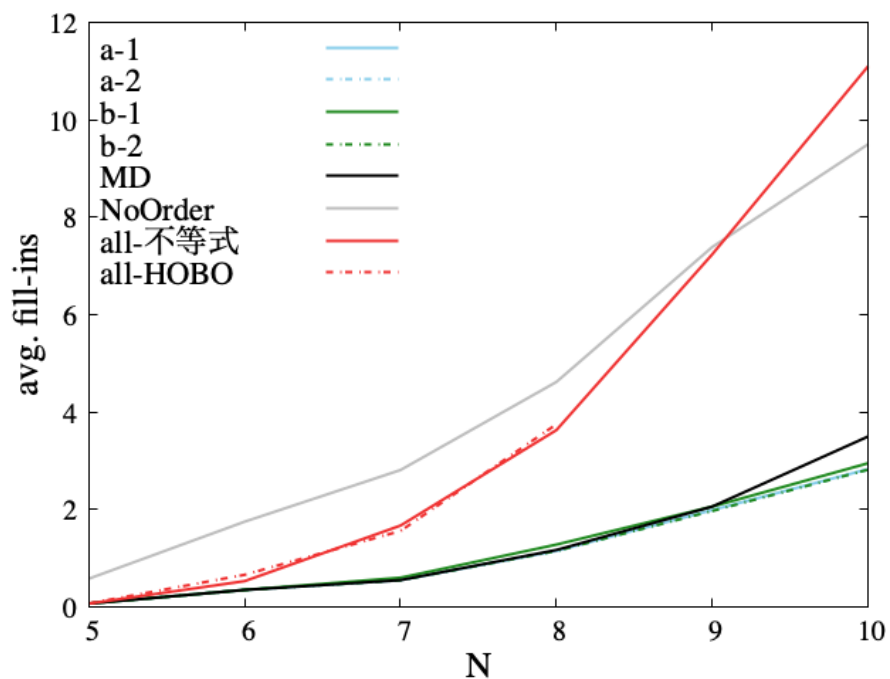


図 5.9 各オーダリング手法における平均フィルイン数

の 10^4 倍の計算時間が必要になる．ハイブリッドオーダリングは、何度も QUBO を生成する必要があるため QMF のボトルネックである QUBO の生成時間が課題となる．

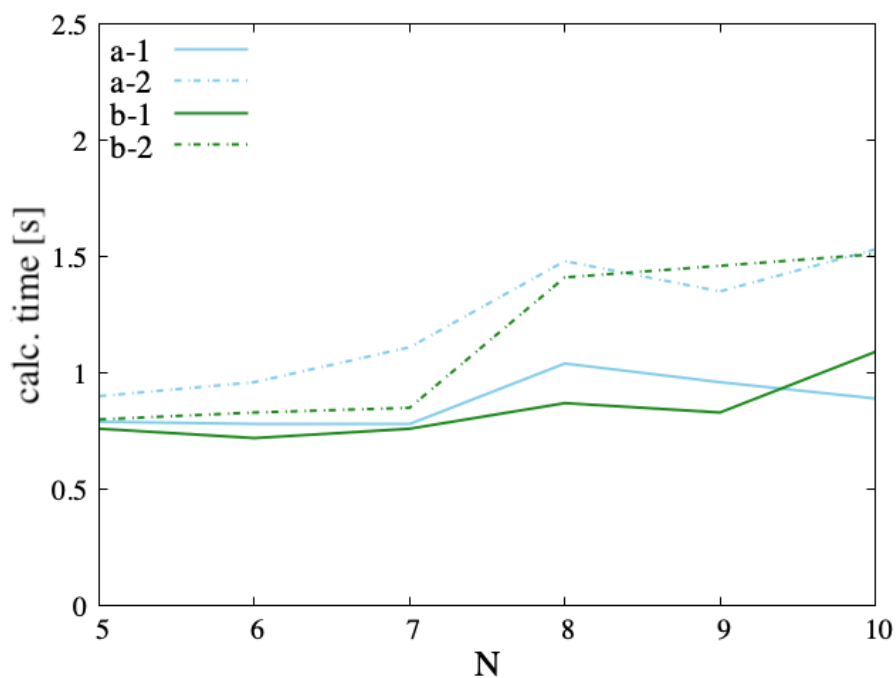


図 5.10 各オーダリング手法における平均計算時間

第 6 章

結論

6.1 まとめ

量子コンピューティング技術が実現されて以降、これらの技術を効果的に利用する方法を提案することの重要性が高まっている。そこで、本研究では、QA を利用したオーダリング手法を例に QA に適した複雑な制約を持つ問題に対する定式化手法の提案と評価結果を示した。さらに、現状の QA マシンを有効に利用するための問題緩和手法や従来のアルゴリズムと組み合わせたハイブリッド手法についても議論した。本研究の成果を以下に再掲する。

- QA を利用した疎行列のオーダリング手法である QMF を QUBO 定式化し、これにより従来手法よりも少ないフィルイン数の解が得られることがあることを示した
- 複雑な制約を持つ問題を、QA に適した QUBO に定式化する手法として、次数削減と不等式を利用した 2 つの定式化手法を評価した
- QA におけるパラメータチューニングの重要性を実験により定量的に示した
- QA を効果的に利用する手法として、問題緩和と従来アルゴリズムとのハイブリッドオーダリングを提案し、評価結果を示した

これらの成果は、QA の応用研究の発展の一助となるものである。

6.2 今後の課題

今後 QA や QIA マシンが発展し、扱える量子ビット数の増加や得られる解の質が向上した際に、本研究で提案した手法をより大きな問題で評価する必要がある。QA を利用して実用規模の問題を解く場合、計算に必要な QUBO 行列の生成時間がボトルネックになることが考えられる。一般的に、QA で問題を解く時間は問題サイズが増加してもほぼ変化しないとされているが、QUBO 行列の生成は従来のコンピュータを利用して行われるため、問題サイズが大きくなると生成時間も指数的に増加する。この課題に対して、事前に生成した QUBO 行列に微修正を加えて使い続ける研究 [54] もされているが、QUBO 行列自体を高速生成することが必要になっていくと考えられる。QUBO 行列の高速生成には、並列処理な

ど従来研究されてきた高速化アルゴリズムが有効になると考えている。

また、ハイブリッドオーダリングに関して、本研究では従来手法の MD 法と QMF のオーダリングを検討したが、その他のオーダリング手法とのハイブリッド手法も検討可能である。例えば、RCM 法は、開始頂点にオーダリングの有効性が依存することや、レベル数が多くなり同期オーバーヘッドが大きくなること、頂点数が多くなると計算時間が大きくなることが課題として挙げられる。そこで、RCM のレベル分けの処理を QA で実施することにより、開始点への依存性を低減し、少ないレベル数と計算時間でレベル分けができる可能性がある。また、一般的に並列処理向けオーダリングでは、並列に処理できるレベル数が少ないほど同期処理の時間が短く、また、incompatible node（開始頂点（図 3.4 では右下の黒で塗りつぶされた頂点）と同じレベルの頂点）の数やオーダリング後の行列のプロファイルが小さいほど収束性が良いため良いオーダリングと評価される。このような評価基準に沿ったレベル分けを QA により実施し、より収束性の高いオーダリングを RCM 法により得ることができる可能性もある。以上のように、従来利用されてきたオーダリング手法の欠点の補完や、より良いオーダリングを得るために QA を活用することが期待される。

謝辞

本研究にあたり，学部から4年にわたり指導教員としてご指導いただいた鈴木智博教授に深く感謝申し上げます。また，学部，修士も含め長い間，副指導教員を引き受けてくださり，的確なご指摘，ご指導いただいた，岩沼宏治教授，鍋島英知准教授に感謝申し上げます。中間審査や公聴会において貴重なご意見いただいた，伊藤一帆教授，高橋正和教授，服部元信教授に感謝申し上げます。加えて，3年にわたるゼミへの参加をご快諾いただき，本分野の研究に関する有益な情報やご意見をいただいた，東北大学/東京工業大学大関真之教授，大関研究室の皆様感謝申し上げます。また，本研究において無料でQIAマシンを利用させていただき，多くの実験にご協力いただいた Fixstars Amplify 社に感謝申し上げます。本研究は JST 科学技術イノベーション創出に向けた大学フェロシップ創設事業 JPMJFS2117 の支援を受けたものである。

参考文献

- [1] G. E. Moore: Cramming More Components onto Integrated Circuits, Proceedings of the IEEE, vol. 86, no. 1, pp. 82–85, 1998.
- [2] 西森秀稔, 大関真之: 量子アニーリングの基礎, 共立出版, 東京, 2018.
- [3] 理化学研究所: ハロー! スパコン富岳版 第5回これが「富岳」だ, <https://www.r-ccs.riken.jp/intro-hpc/hello-sc-fugaku/05.html>, (2023-11-28).
- [4] 理化学研究所: 量子コンピュータを利用できる「量子計算クラウドサービス」開始 - 国産超伝導量子コンピュータ初号機の公開 -, https://www.riken.jp/pr/news/2023/20230324_1/, (2023-11-29).
- [5] T. Kadowaki and H. Nishimori: Quantum annealing in the transverse Ising model, Phys. Rev. E, vol. 58, no. 5, pp. 5355–5363, 1998.
- [6] 小見山朋子: 量子アニーリングによる疎行列直接解法向けオーダリング手法, 山梨大学卒業論文, 2021.
- [7] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno: 20k-spin Ising chip for combinatorial optimization problem with CMOS annealing, IEEE Journal of Solid-State Circuits, vol. 51, no. 1, pp. 303–309, 2015.
- [8] A. Lucas: Ising formulations for many NP problems, Frontiers in Physics, 2014.
- [9] M. Schuld, F. Petruccione 著, 大関真之 監訳, 荒井俊太, 篠島匠人, 高橋茶子, 御手洗光祐, 山城悠 訳: 量子コンピュータによる機械学習, 共立出版, 2020.
- [10] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi: Optimization by Simulated Annealing, SCIENCE, vol. 220, pp. 671–680, 1983.
- [11] S.Boxio, T.F.Ronnow, S.V.Isakov, Z.Wang, D.Wecker, D.A.Lidar, J.M.martinis and M.troyer: Evidence for quantum annealing with more than one hundred qubits, Nature physics, vol. 10, no. 3, pp. 218–224, 2014.
- [12] R.Hamerly, T.Inagaki, P.L.McMahon, D.Venturelli, A. Marandi, T. Onra, E. Ng, C. Langrock, K. Inaba, T. Honjo et al. : Experimental investigation of performance differences between coherent Ising machines and a quantum annealer, Science advances, vol. 5, no. 5, eaau0823, 2019.
- [13] F. Arute, et.al.: Quantum supremacy using a programmable superconducting processor, Nature, vol. 574, pp. 505–510, 2019.

- [14] Fixstars: QUANTUM COMPUTING SOLUTIONS, <https://quantum.fixstars.com/techresouces/annealing-method/programming/index.html> (2021-02-16).
- [15] S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, K. Takemoto and H. Tamura: Digital Annealer for High-Speed Solving of Combinatorial optimization Problems and Its Applications, 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 667–672, 2020.
- [16] K. Yonaga, M. J. Miyama, M. Ohzeki: Solving Inequality-Constrained Binary Optimization Problems on Quantum Annealer, arXiv:2012.06119v1, 2020.
- [17] 谷池悠太, 戸川望: シフト法とスピン追加法を効率的に組み合わせたイジングモデルのビット削減手法の評価, 情報処理学会研究報告, vol. 2021-QS-4, no. 15, 2021.
- [18] M. Kuramata, R. Katsuki, K. Nakata: Solving large break minimization problems in a mirrored double round-robin tournament using quantum annealing, PLOS ONE, vol. 17, no. 4, e0266846, 2022.
- [19] K. Nakano, S. Tsukiyama, Y. Ito, T. Yazane, J. Yano, T. Kato, S. Ozaki, R. Mori, R. Katsuki: Dual-Matrix Domain-Wall: A Novel Technique for Generating Permutations by QUBO and Ising Models with Quadratic Sizes, Technologies, (preprint) <https://doi.org/10.3390/technologies11050143>, 2023.
- [20] 上島雄高, 田中宗: ルールベース変数固定された QUBO のイジングマシン上の評価, 情報処理学会研究報告, vol. 2023-QS-10, no. 4, 2023.
- [21] 矢田部彰宏: クラスタ状に分布する都市に関する巡回セールスマン問題の QUBO 行列による問題分割手法, 情報処理学会研究報告, vol. 2023-QS-8, no. 6, 2023.
- [22] 奥山拓哉, 吉村地尋, 林 真人, 田中 咲, 山岡雅直: イジング計算機に向けたグラフ埋め込みアルゴリズム, 信学技法, vol. 116, no. 116, COMP2016-11, pp. 97-103, 2016.
- [23] O. Kroger: Engineering Minimum Degree Node Ordering Algorithm, Ruprecht-Karls-Universitat Heidelberg, <https://schulzchristian.github.io/thesis/thesis01eKroeger.pdf>, 2021.
- [24] T. A. Davis: Direct Methods for Sparse Linear Systems, siam, 2006.
- [25] M. Yannakakis: Computing the Minimum Fill-In is NP-Complete. 2. <http://dx.doi.org/10.1137/0602010>, 1981.
- [26] F. V. Fomin, D. Kratsch, I. Todinca, Y. Villanger: Exact Algorithms for Treewidth and Minimum Fill-In, vol. 38, no. 3, pp. 1058–1079, <http://dx.doi.org/10.1137/050643350>, Society for Industrial and Applied Mathematics, 2008.
- [27] D. J. Rose, R. E. Tarjan, G. S. Lueker: Algorithmic Aspects of Vertex Elimination on Graphs, vol. 5, no. 2, pp. 266–283. <http://dx.doi.org/10.1137/0205021>, Society for Industrial and Applied Mathematics, 1976.
- [28] E. Dahlhaus: Minimal elimination ordering for graphs of bounded degree, vol.

- 116, no. 1, pp. 127–143. [http://dx.doi.org/10.1016/S0166-218X\(00\)00331-0](http://dx.doi.org/10.1016/S0166-218X(00)00331-0). - DOI10.1016/S0166-218X(00)00331-0, 2002.
- [29] W. F. Tinney, J. W. Walker: Direct solutions of sparse network equations by optimally ordered triangular factorization, vol. 55, no. 11, pp. 1801–1809. <http://dx.doi.org/10.1109/PROC.1967.6011>, Proceedings of the IEEE, 1967.
- [30] I. S. Duff, A. M. Erisman and J. K. Reid: chapter in Direct methods for sparse matrices, UK: OXFORD, 2017.
- [31] A. George, J. W. Liu: The Evolution of the Minimum Degree Ordering Algorithm, vol. 31, no. 1, pp. 1–19. <http://dx.doi.org/10.1137/1031001>, Society for Industrial and Applied Mathematics, 1989.
- [32] P. R. Amestoy, T. A. Davis, I. S. Duff: An Approximate Minimum Degree Ordering Algorithm, vol. 17, no. 4, pp. 886–905, <http://dx.doi.org/10.1137/S0895479894278952>, Society for Industrial and Applied Mathematics, 1996.
- [33] A. George: Nested Dissection of a Regular Finite Element Mesh, vol. 10, no. 2, pp. 345–363. <http://dx.doi.org/10.1137/0710032>, Society for Industrial and Applied Mathematics, 1973.
- [34] A. George, J. W. H. Liu: An Automatic Nested Dissection Algorithm for Irregular Finite Element Problems, vol. 15, no. 5, pp. 1053–1069. <http://dx.doi.org/10.1137/0715069>, Society for Industrial and Applied Mathematics, 1978.
- [35] E. Cuthill, J. McKee: Reducing the Bandwidth of Sparse Symmetric Matrices. Proceedings of the 1969 24th National Conference. New York, NY, USA: Association for Computing Machinery. pp. 157–172, 1969.
- [36] A. George, J. W. Liu: Computer Solution of Large Sparse Positive Definite. Prentice Hall Professional Technical Reference, ISBN 0131652745, 1981.
- [37] 俵谷健太郎, 横川三津夫: 色数を抑えた改良 Reverse Cuthill-McKee 法による線形ソルバの並列化について, 情報処理学会第研究報告, vol. 2017-HPC-159, no. 3, pp.1–6, 2017.
- [38] G. H. Golub, and C. F. V. Loan: Matrix Computations (4th edition), pp. 602-604, JJU Press, 2013.
- [39] 中島研吾, 大島聡史, 林雅江: 科学技術計算におけるマルチコアプログラミング入門 第II部オーダリング, <https://www.cc.u-tokyo.ac.jp/events/lectures/18/omp-02.pdf>, <https://www.cc.u-tokyo.ac.jp/events/lectures/18/omp-03.pdf>, 2011.
- [40] 中島研吾: 拡張型 ELL 行列格納手法に基づくメニィコア向け疎行列ソルバー, 情報処理学会研究報告, vol. 2014-ARC-213, no. 3, vol. 2014-HPC-147, no. 3, pp.1–8, 2014.
- [41] V. Kolmogorov and R. Zabih: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26, pp. 147–159, 2004.

- [42] D. Freedman and P. Drineas: Energy Minimization via Graph Cuts: Settling What is Possible, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 05), vol.2, pp. 939-946, 2005.
- [43] H. Ishikawa: Transformation of General Binary MRF Minimization to the First-Order Case', IEEE Transactions on Pattern Analysis and Machine Intelligence 33, pp.1234–1249, 2011.
- [44] N. Dattani: Quadraticization in discrete optimization and quantum mechanics, Quantum Physics, <http://arxiv.org/abs/1901.04405>, 2019.
- [45] 川上蒼馬, 武笠陽介, 鮑思雅, 巴德瑪, 新井淳也, 八木哲志, 寺本純司, 戸川望: イジングマシンによる制約付きグラフ彩色問題の彩色数最小化手法, 情報処理学会研究報告, vol. 2022-QS-05, no. 2, 2022.
- [46] H. N. Djidjev, Quantum Annealing with Inequality Constraints: The Set Cover Problem, Adv Quantum Technol., vol. 6, issue. 11, 2300104, 2023.
- [47] C. M. Li and F. Manya: The Handbook of Satisfiability (2nd. ed.). Frontiers in Artificial Intelligence and Applications, vol. 336. IOS Press, Chapter 23, 2021.
- [48] A. Mandal, A. Roy, S. Upadhyay, and H. Ushijima-Mwesigwa: Compressed Quadraticization of Higher Order Binary Optimization Problems. arXiv preprint, <https://doi.org/10.48550/arXiv.2001.00658>, 2020.
- [49] 宋剛秀, 番原睦則, 田村直之, 鍋島英知: SAT ソルバーの最新動向と利用技術, コンピュータソフトウェア, vol. 35, no. 4, pp. 72–92, https://www.jstage.jst.go.jp/article/jssst/35/4/35_72/_pdf, 2018.
- [50] Naoki Ide, Tetsuya Asayama, Hiroshi Ueno, Masayuki Ozeki: Maximum Likelihood Channel Decoding with Quantum Annealing Machine,<https://doi.org/10.48550/arXiv.2007.08689>.
- [51] K. Tamura, T. Shirai, H. Katsura, S. Tanaka and N. Togawa: Performance Comparison of Typical Binary-Integer Encodings in an Ising Machine, in IEEE Access, vol. 9, pp. 81032–81039, 2021.
- [52] M. Zaman, K. Tanahashi, S. Tanaka: PyQUBO: Python Library for Mapping Combinatorial Optimization Problems to QUBO Form, IEEE Transactions on Computers, vol. 71, no. 4, pp. 838–850, 2022.
- [53] IBM: IBM ILOG CPLEX Optimization Studio, <https://www.ibm.com/jp-ja/products/ilog-cplex-optimization-studio>, (2023/12/18).
- [54] 土井龍太郎, 鷹野美美代: モデル差分を用いた QUBO 行列の高速生成, 情報処理学会研究報告, vol. 2023-QS-10, no. 3, 2023.
- [55] N. Chancellor: Domain wall encoding of discrete variables for quantum annealing and QAOA, Quantum Sci. Technol. vol. 4, 045004, 2019.
- [56] J. Chen, T. Stollenwerk, N. Chancellor: Performance of Domain-Wall Encoding

for Quantum Annealing, IEEE Transactions on Quantum Engineering, vol. 2, pp. 1–14, 2021.

- [57] 大野乾太郎, 戸川望: 変数の順序付による二値最適化の線形化, 情報処理学会研究報告, vol. 2023-QS-8, no. 2, 2023.
- [58] 株式会社 NTT データ数理システム: Nuorium Optimizer C++ SIMPLE 例題集 V25 2.6 集合被覆問題, <https://www.msi.co.jp/solution/nuopt/docs/examples/html/02-06-00.html>, (2023-6-20).
- [59] 小見山朋子: アニーリングマシンのための時間制約付き経路最適化問題の QUBO 定式化, 山梨大学卒業論文, 2022.

関連文献

- [a] 小見山朋子, 鈴木智博: 疎行列のフィルイン削減オーダリングにおける複雑な制約の定式化, 情報処理学会論文誌, 2024 (審査中).
- [b] Tomoko Komiyama, Tomohiro Suzuki: Problem Relaxation Methods for Quantum Minimum Fill-in Algorithm, *IEEE Access*, vol. 11, pp. 14424-14431, 2023.
- [c] T. Komiyama and T. Suzuki: Sparse matrix ordering method with a quantum annealing approach and its parameter tuning, *Proceedings of IEEE 14th International Symposium on Embedded Multicore/Many-core SoCs (MCSoc-21), Special Session on Auto-Tuning for Multicore and GPU (ATMG)*, pp. 258–264, Dec. 2021, 10.1109/MCSoc51149.
- [d] Tomoko Komiyama, Tomohiro Suzuki: QUBO formulation using inequalities for problems with complex constraints, *Proceedings of The International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia)*, 2024, Nagoya, Japan, ACM, New York, NY. USA, 7 pages. <https://doi.org/10.1145/3635035.3635042f>.

付録 A

Domain-wall encoding による QMF の QUBO 定式化

4 章では QMF を One-hot encoding と呼ばれる定式化手法で QUBO に定式化した。One-hot encoding は，図 A.1 左に示すように， N 個の変数のうち 1 つだけを 1，それ以外を 0 とすることで N 種類の数表現する手法である。ただし，この One-hot encoding と不等式の QUBO 変換手法の Onehot encoding は別物であることに注意が必要である。一方，近年注目されている定式化手法として Domain-wall encoding がある [55, 56]。この手法は，図 A.1 右に示すように， $N - 1$ 個の変数のうち 0 と 1 の境目を 1 箇所のみ生成することで N 種類の数表現する。

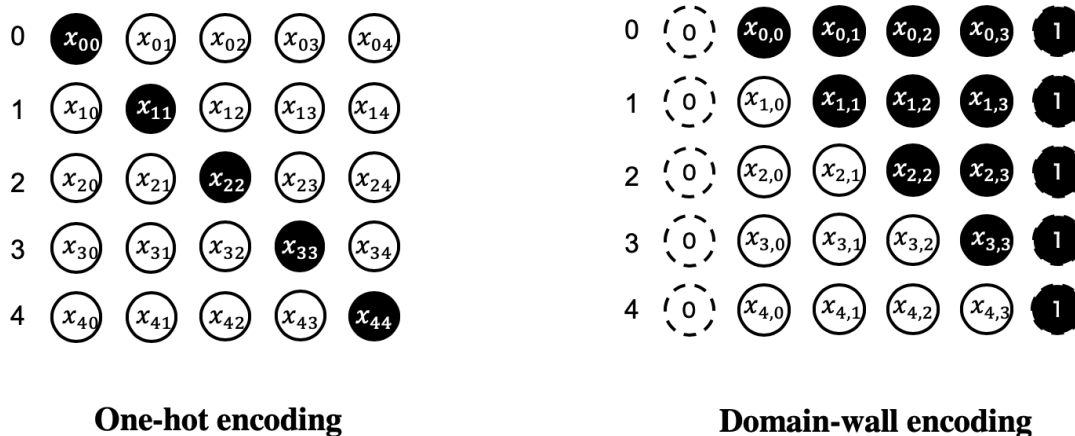


図 A.1 One-hot encoding と Domain-wall encoding

ここで，One-hot encoding と Domain-wall encoding の比較を表 A.1 に示す。この表 A.1 から，One-hot encoding よりも Domain-wall encoding の方が変数と相互作用（変数間の接続数）が少なくなることがわかる。

表 A.1 One-hot encoding と Domain-wall encoding の比較 [55]

変換手法	量子ビット数	相互作用数	グラフ構造
One-hot	m	$m(m-1)$	完全グラフ
Domain-wall	$m-1$	$m-2$	線形グラフ

A.1 QUBO 定式化

頂点の消去順を表現する二値変数 $x_{n,s} \in \{0, 1\}$ と、頂点 u と v 間の辺の存在を表現する二値変数 $y_{\langle u,v \rangle} \in \{0, 1\}$ を用意する. $x_{n,s-1}$ と $x_{n,s}$ の間に wall (0 と 1 の境界) があるとき、頂点 n を第 s ステップに消去することを表し、 $y_{\langle u,v \rangle} = 1$ は、頂点 u と頂点 v の間の無向辺が存在する (非存在辺であれば追加される)、つまり、分解後の行列において (u, v) 要素または (v, u) 要素が非零であることを表す. この変数設定の例を図 A.2 に示す. 図左は行列の消去グラフ、図中央は頂点 $0 \sim 4$ の消去ステップを表す変数集合 \mathbf{x} 、図右は各頂点間辺の有無を表す変数集合 \mathbf{y} である. また、この図では白抜きの変数は 0、黒で塗りつぶした変数は 1 を表す.

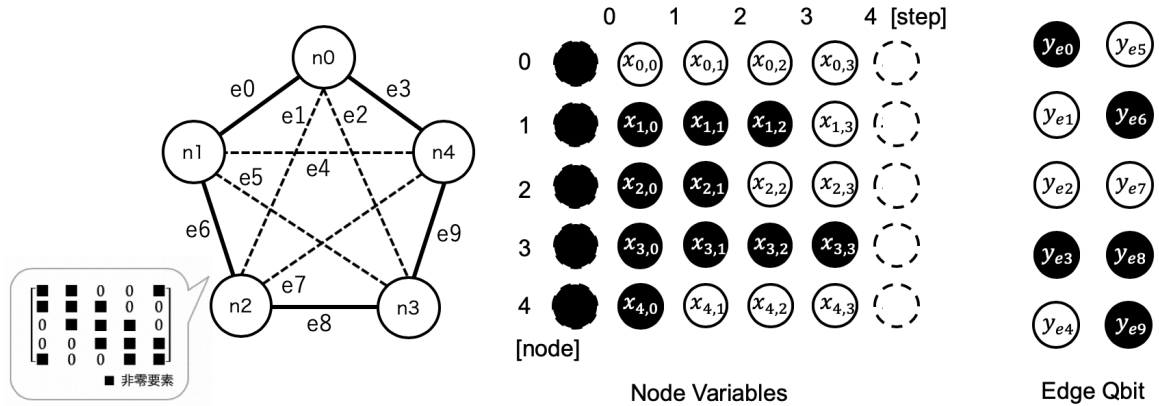


図 A.2 QMF の変数設定例 (Domain-wall encoding)

目的関数. QMF は追加される辺の総数を最小化する

目的関数は、One-hot encoding による定式化の目的関数 (4.1) と同様である.

制約 1. 全ての頂点が各ステップに 1 頂点ずつ消去される

この制約を満たすとき最小値 $-N$ となる関数を次式で定義する.

$$H_{D1} = - \sum_n^{N-1} \sum_s^{N-2} (x_{n,s} - x_{n,s+1})^2 + \sum_s^{N-2} \sum_n^{N-1} \sum_{n^*}^{N-1} (x_{n,s} - x_{n,s+1})(x_{n^*,s} - x_{n^*,s+1}) \quad (\text{A.1})$$

この関数の第 1 項目は図 A.3(a) のように、頂点 n を表す変数 $x_{n,s}$ のどこか 1 箇所に

wall (0 と 1 の境界) があるとき、つまり、頂点 n が 1 度だけ消去される時最小値 $-N$ をとる。第 2 項目は図 A.3(b) のように、頂点 n 表す変数 $x_{n,s}$ と頂点 n^* 表す変数 $x_{n^*,s}$ の同じ箇所に wall (0 と 1 の境界) がないとき、つまり、各消去ステップ s において頂点が 1 つだけ消去される時最小値 0 をとる。

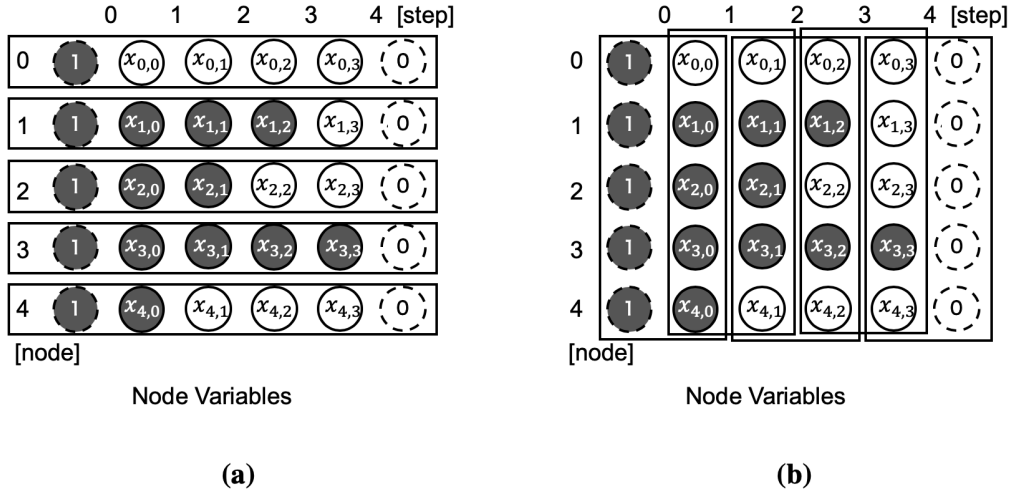


図 A.3 制約 1 の変数例 (Domain-wall encoding)

制約 2. 頂点の消去により必要な辺が追加される

各ステップにおいて、消去対象の頂点に接続のある頂点間が全結合となるように辺の追加を行う (図 4.3 参照)。ここで辺の追加は次の条件により発生する。

- 頂点 n を s ステップ目で消去する ($x_{n,s} - x_{n,s+1} = 1$)
- 頂点 u が s ステップ目よりも前のステップで消去されていない ($x_{u,s} = 1$)
- 頂点 v が s ステップ目よりも前のステップで消去されていない ($x_{v,s} = 1$)
- 消去する頂点 n と頂点 u を結ぶ辺が存在する ($y_{\langle n,u \rangle} = 1$)
- 消去する頂点 n と頂点 v を結ぶ辺が存在する ($y_{\langle n,v \rangle} = 1$)

以上の条件が全て満たされた時に、頂点 u と頂点 v を接続する辺を追加する ($y_{\langle u,v \rangle} = 1$)。以上の制約を満たす制約は各条件の積の形式で定義される。

$$H_{D2} = \sum_{(u,v) \in E} \sum_{n \notin \{u,v\}} \sum_{s=0}^{N-2} (1 - y_{\langle u,v \rangle}) (x_{n,s} - x_{n,s+1}) x_{v,s} y_{\langle n,u \rangle} y_{\langle n,v \rangle} \quad (\text{A.2})$$

この関数は 6 次式であるため、4 章で議論した手法により QUBO に変換する。

高次式の次数削減

QMF の制約関数 (A.2) は展開すると 5~6 次の項を含む多項式となる。一方、次の補助変数 \mathbf{b} を導入すると 6 次の単項式の差で表すことができる。

$$b_{\langle u,v \rangle} = 1 - y_{\langle u,v \rangle} \quad (\text{A.3})$$

$$H_1^{D2a} = \sum_{\langle u,v \rangle \in \bar{E}} (b_{\langle u,v \rangle} + y_{\langle u,v \rangle} - 1)^2 \quad (\text{A.4})$$

$$\begin{aligned} H_2^{D2a} &= \sum_{(u,v) \in E} \sum_{n \notin \{u,v\}} \sum_{s=0}^{N-2} b_{\langle u,v \rangle} (x_{n,s} - x_{n,s+1}) x_{u,s} x_{v,s} y_{\langle n,u \rangle} y_{\langle n,v \rangle} \\ &= \sum_{(u,v) \in E} \sum_{n \notin \{u,v\}} \sum_{s=0}^{N-2} b_{\langle u,v \rangle} x_{n,s} x_{u,s} x_{v,s} y_{\langle n,u \rangle} y_{\langle n,v \rangle} \\ &\quad - b_{\langle u,v \rangle} x_{n,s+1} x_{u,s} x_{v,s} y_{\langle n,u \rangle} y_{\langle n,v \rangle} \end{aligned} \quad (\text{A.5})$$

この関数の括弧内の第 1 項目を Ishikawa 法で次数削減すると $2N(N-1)\bar{E}$ の補助変数, 第 2 項目を KZFD 法で次数削減すると $N(N-1)\bar{E}$ の補助変数で次数削減可能であり, 補助変数 \mathbf{b} の分も加えても $4N(N-1)\bar{E}$ 程度の補助変数で QUBO に変換できる. ただし, 疎行列の場合, 式 (A.5) は One-hot encoding の式 (4.10) と同様に $O(N^4)$ である.

不等式の利用

制約 2 を主制約, 制約 2 のそれぞれの条件を副制約とすると不等式で定式化できる.

$$(1 - y_{\langle u,v \rangle}) + (x_{n,s} - x_{n,s+1}) + x_{u,s} + x_{v,s} + y_{\langle n,u \rangle} + y_{\langle n,v \rangle} \leq 5 \quad (\text{A.6})$$

この不等式の左辺は, 5 つの副制約がすべて満たされるが辺 $y_{\langle u,v \rangle}$ が追加されないとき最大値 6 をとり, そうでないときは 0 から 5 の間の値をとる. したがって, この不等式が成り立つとき, 主制約が満たされることを表す. さらに, この不等式はスラック変数 \mathbf{d} を導入することで, QUBO に変換することができる.

$$\begin{aligned} H_{D2b} &= \sum_{(u,v) \in E} \sum_{n \notin \{u,v\}} \sum_{s=0}^{N-2} \left\{ (1 - y_{\langle u,v \rangle}) + (x_{n,s} - x_{n,s+1}) + x_{u,s} + x_{v,s} \right. \\ &\quad \left. + y_{\langle n,u \rangle} + y_{\langle n,v \rangle} - \sum_{l=0}^4 d_l \right\}^2 \end{aligned} \quad (\text{A.7})$$

この関数は, この制約が満たされていれば 0 をとり, そうでなければ 1 をとる.

次数削減を利用した QUBO は次式である.

$$H_{Da} = H_{cost} + \alpha H_{D1} + \beta H_1^{D2a} + \gamma H_2^{D2a} \quad (\text{A.8})$$

α, β, γ は 4 章の式 (4.31) と同様に設定した. また, 不等式を利用した QUBO は次式で

ある.

$$H_{Db} = H_{cost} + \sigma H_{D1} + \epsilon H_{D2b} \quad (\text{A.9})$$

σ と ϵ の値は 4 章の式 (4.34) と同様に設定した. 制約が満たされているかどうかは, 次数削減を利用した QUBO の場合は制約関数 (A.1), (A.4), (A.5) 不等式を利用した QUBO の場合は制約関数 (A.1), (A.7) に得られた解 \mathbf{x} を代入し, エネルギー値が 0 になっていることを確認することで判定した.

A.2 評価

One-hot encoding と Domain-wall encoding による QMF を比較する. まず, 図 A.4 から, One-hot encoding よりも Domain-wall encoding の方が計算に必要な変数が少なく, 特に, Domain-wall encoding と次数削減を利用した QUBO が最も変数が少なくなることがわかる. これは, Domain-wall encoding と次数削減を利用した QUBO が最も補助変数の追加が少ないためである.

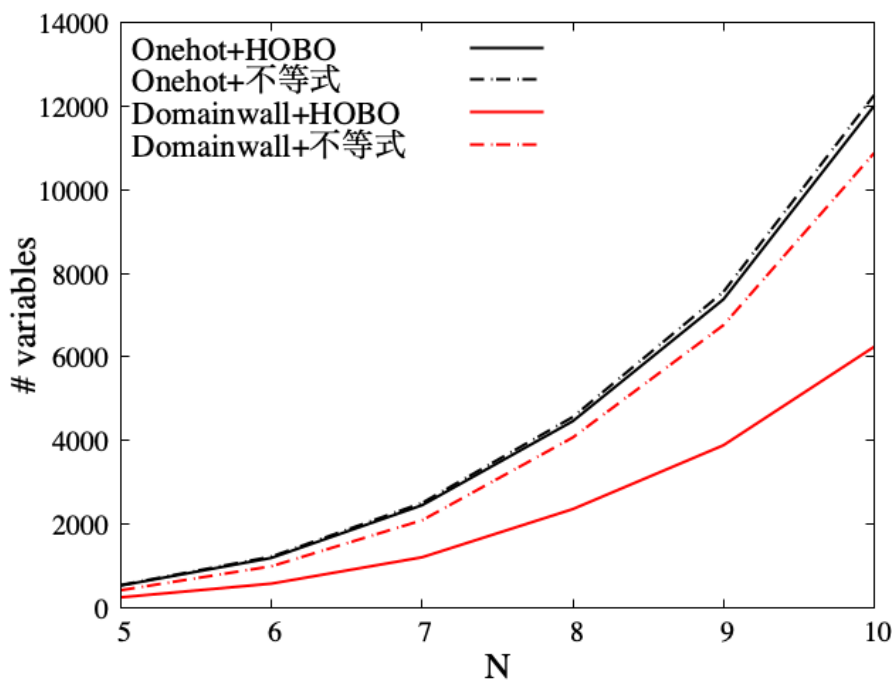


図 A.4 One-hot encoding と Domain-wall encoding による QMF の変数の数

また, 図 A.5 に各 QUBO の変数の接続数を示す. ここから, 全ての QUBO において問題が大きくなるにつれて変数の平均接続数が大きくなることがわかる. また, One-hot encoding よりも Domain-wall encoding, 次数削減を利用した QUBO よりも不等式を利用した QUBO の方が変数の平均接続数が少なくなることがわかる. 4 章で述べた通り, 変数の接続数は少ないほど疎結合グラフをもつ QA/QIA マシンで解が得られやすい. そのため, 現状の QA マシンには Domain-wall+ 不等式が最も適していると考えられる.

さらに, 図 A.6 から, 次数削減を利用した定式化よりも不等式による定式化の方が実行可能解が得やすく, 4 つの QUBO の中では Onehot+ 不等式が最も実行可能解を得やすいこと

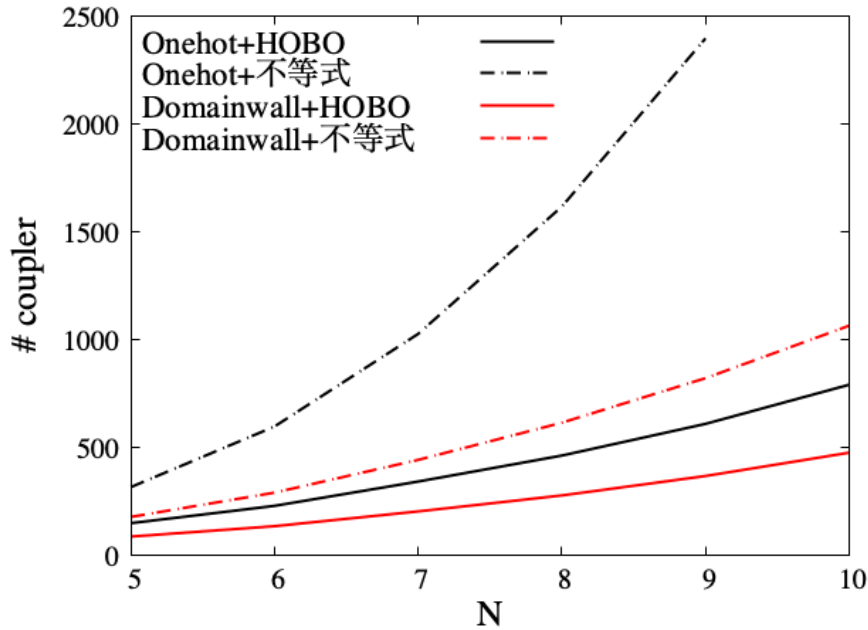


図 A.5 One-hot encoding と Domain-wall encoding による QMF の変数の平均接続数

がわかる。また、問題サイズが 6×6 までと極めて小さい場合を除き、Domain-wall による 2 つの QUBO は One-hot+HOBO よりも実行可能解を得やすい。これは、One-hot+HOBO よりも Domain-wall+HOBO の方が補助変数の制約が少ないためであると考えられる。しかし、図 A.7 から最適解は Domain-wall+ 不等式が他の 3 手法よりも得やすいこともわかる。以上の結果から、次数削減を利用した定式化よりも不等式による定式化、One-hot encoding よりも Domain-wall encoding による定式化の方が質の良い解を得やすいと考える。

A.3 QUBO 行列の解析

前述の通り、QUBO 定式化手法により解の得やすさが異なる。解の得やすさと QUBO の関連性を調査することを目的に、それぞれの QUBO 行列の固有値、特異値、条件数を比較した。先に結論を述べておくと、今回の解析結果からは解の得やすさに依存する要素がわからなかった。QUBO の生成に利用した疎行列は、ランダムに非零要素がある行列と対角に非零要素を集めた行列どちらも結果に大差はなかったため、本節ではランダムに非零要素がある行列の解析結果を示す。

A.3.1 固有値

各 QUBO 行列の最小固有値を図 A.8、最大固有値を図 A.9、固有値範囲を図 A.10 に示す。まず、図 A.8 から、最小固有値は $\text{One-hot+HOBO} < \text{One-hot+ 不等式} < \text{Domain-wall+ 不等式} < \text{Domain-wall+HOBO}$ の順に小さいことがわかる。次に、図 A.9 から、最大固有値は $\text{One-hot+HOBO} > \text{One-hot+ 不等式} > \text{Domain-wall+ 不等式} > \text{Domain-wall+HOBO}$ の順に大きいことがわかる。また、図 A.10 から、固有値範囲は $\text{One-hot+HOBO} > \text{One-hot+}$

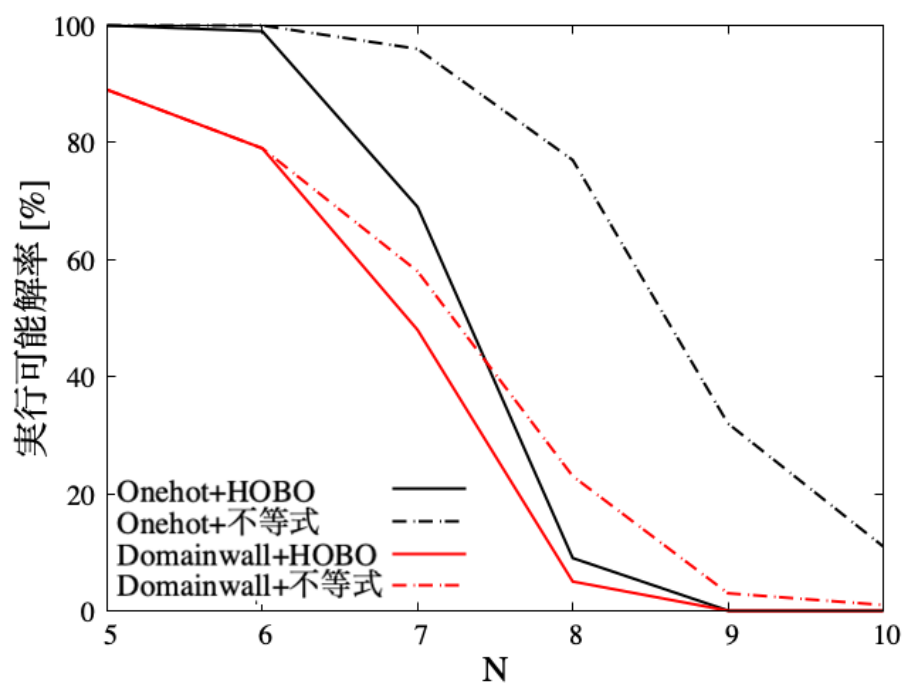


図 A.6 One-hot encoding と Domain-wall encoding の実行可能解が得られた確率の比較

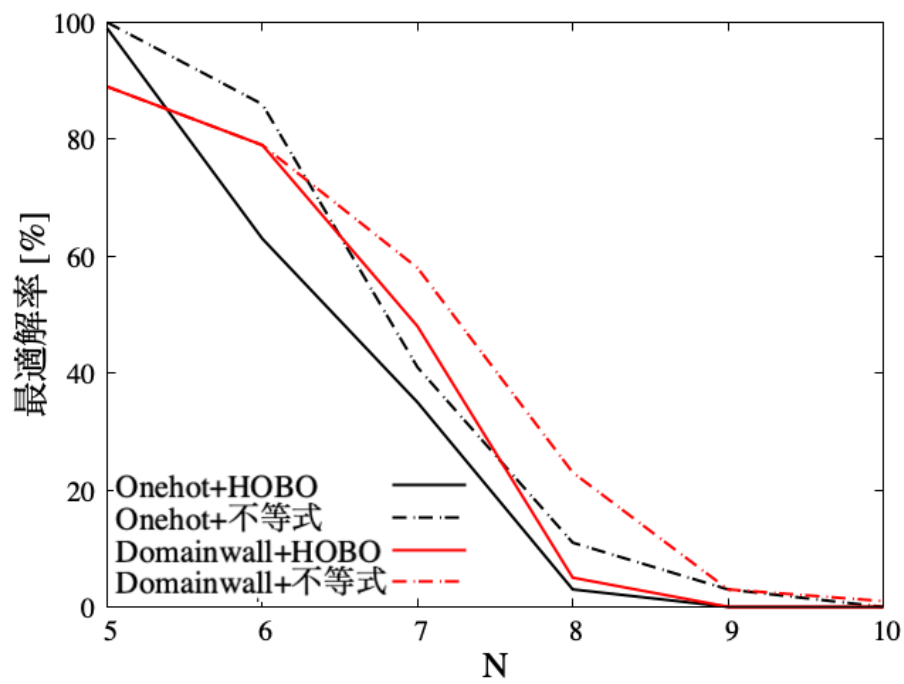


図 A.7 One-hot encoding と Domain-wall encoding の最適解が得られた確率の比較

不等式 > Domain-wall+ 不等式 > Domain-wall+HOBO の順に大きいことがわかる。以上の結果と前述の図 A.6, 図 A.7 の結果から, 問題の解きやすさと QUBO 行列の固有値には関連性はないと考える。

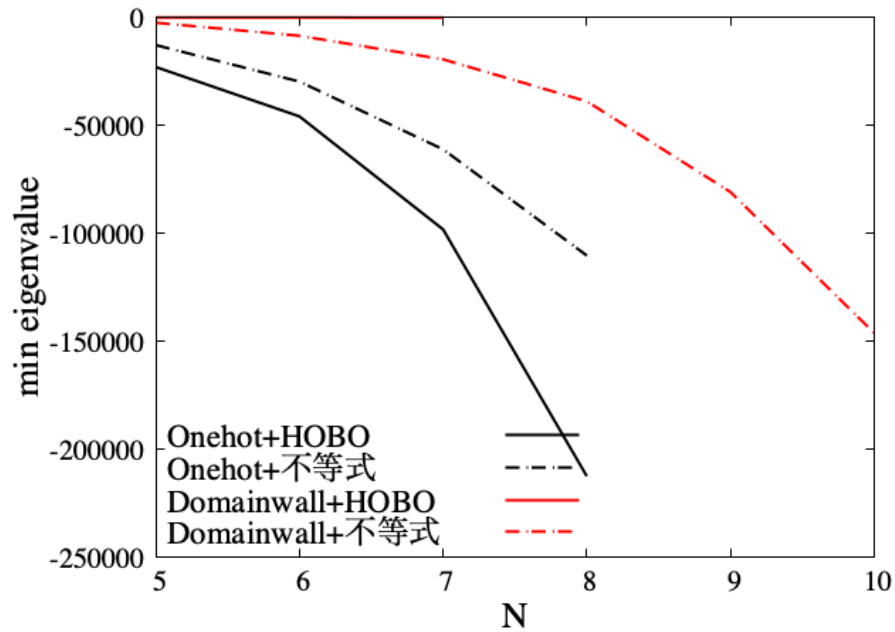


図 A.8 各 QUBO 行列の最小固有値

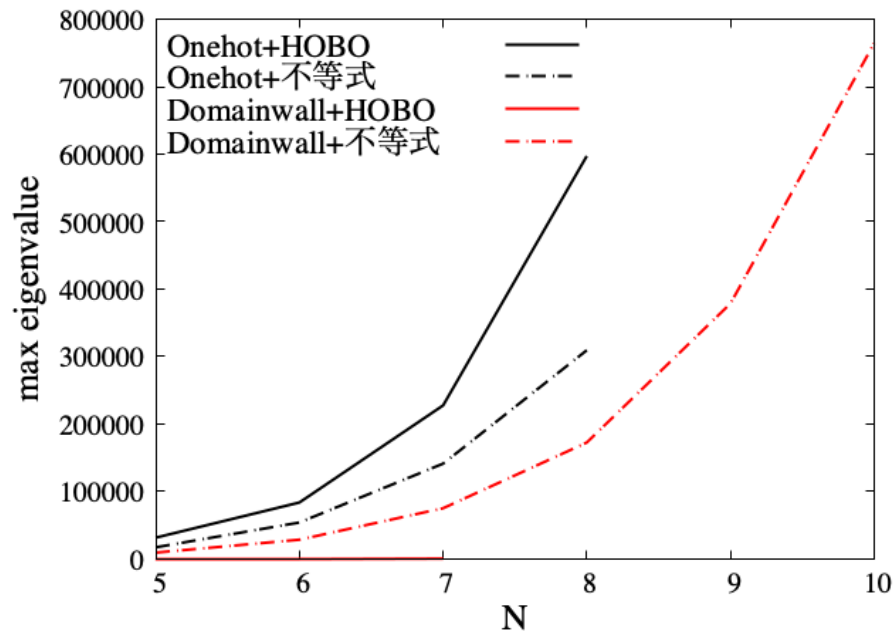


図 A.9 各 QUBO 行列の最大固有値

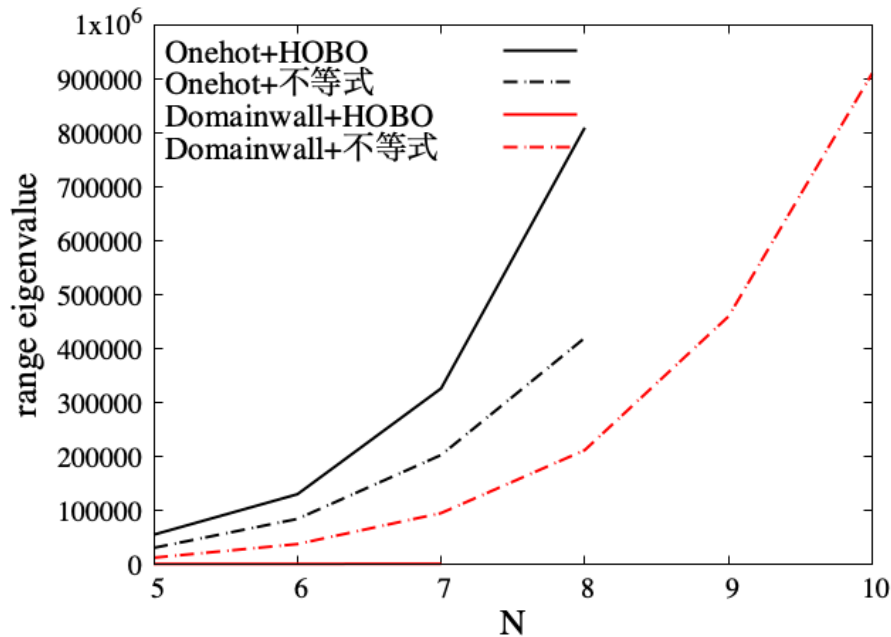


図 A.10 各 QUBO 行列の固有値範囲

A.3.2 特異値値

各 QUBO 行列の最小特異値を図 A.11, 最大特異値を図 A.12, 特異値範囲を図 A.13 に示す. まず, 図 A.11 から, 最小特異値は One-hot+HOBO < One-hot+ 不等式 < Domain-wall+ 不等式 < Domain-wall+HOBO の順に小さいことがわかる. 次に, 図 A.12 から, 最大特異値は One-hot+HOBO > One-hot+ 不等式 > Domain-wall+ 不等式 > Domain-wall+HOBO の順に大きいことがわかる. また, 図 A.13 から, 特異値範囲は One-hot+HOBO > One-hot+ 不等式 > Domain-wall+ 不等式 > Domain-wall+HOBO の順に大きいことがわかる. 以上の結果と前述の図 A.6, 図 A.7 の結果から, 問題の解きやすさと QUBO 行列の特異値には関連性はないと考える.

A.3.3 条件数

各 QUBO 行列の条件数を図 A.14 に示す. ここから, 条件数は One-hot+HOBO > One-hot+ 不等式 > Domain-wall+ 不等式 > Domain-wall+HOBO の順に大きいことがわかる. 前述の図 A.6, 図 A.7 の結果から, 問題の解きやすさと QUBO 行列の条件数には関連性はないと考える.

以上の結果から, QUBO 行列の性質と実行可能解や質の良い解の得やすさの関連性は見えなかった.

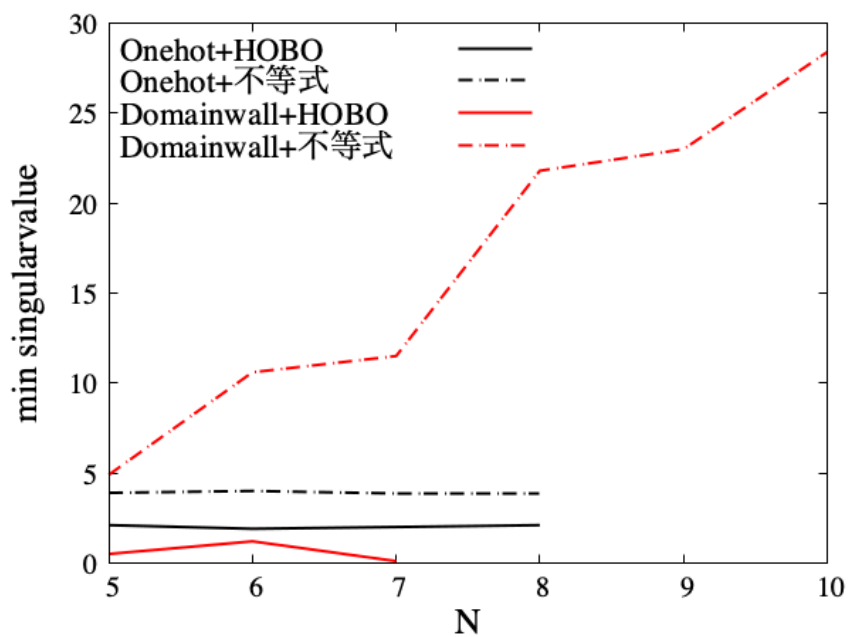


図 A.11 各 QUBO 行列の最小特異値

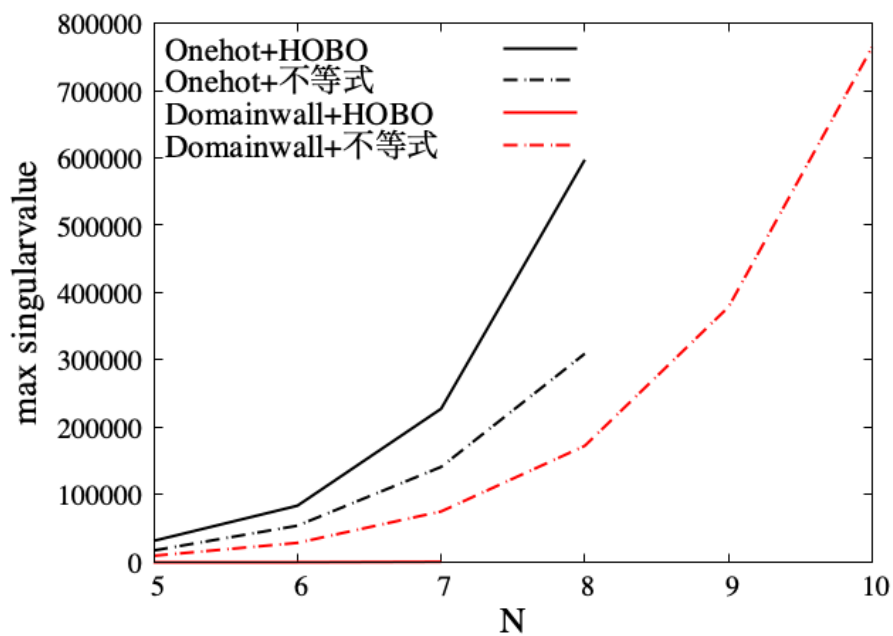


図 A.12 各 QUBO 行列の最大特異値

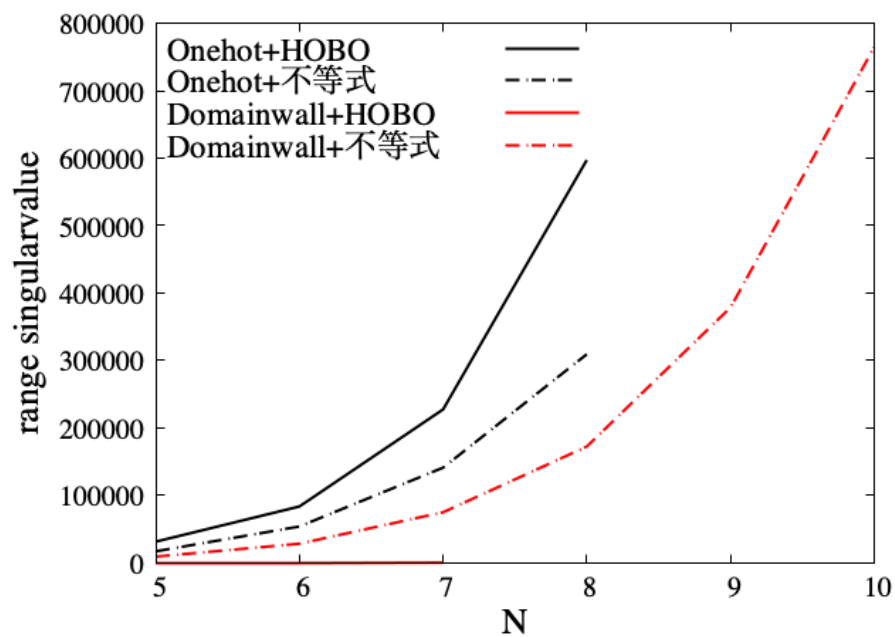


図 A.13 各 QUBO 行列の特異値範囲

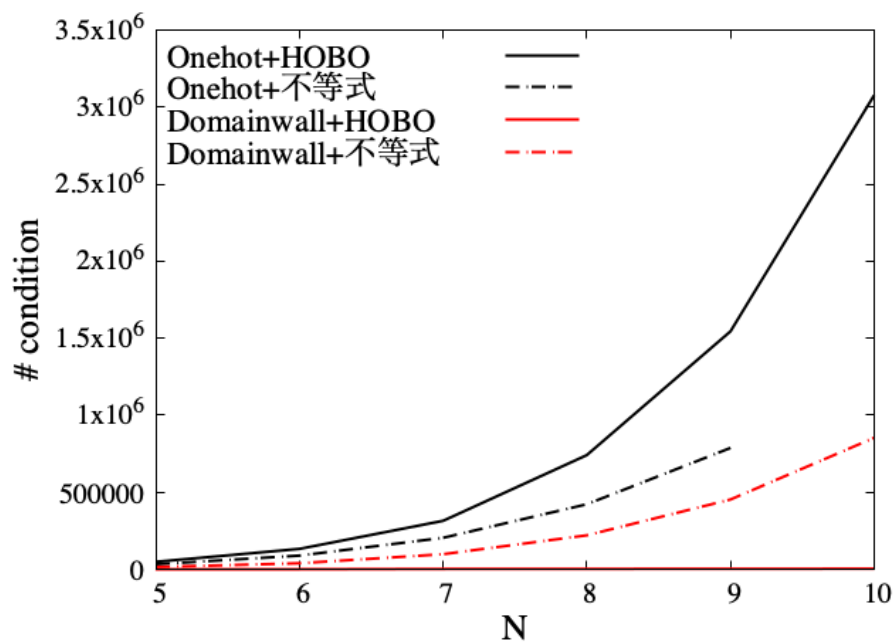


図 A.14 各 QUBO 行列の条件数

付録 B

不等式の QUBO 変換手法の比較

4 章では, QA では不等式をそのまま解くことができず, QUBO に変換するなどの工夫が必要であることを述べた. さらに, MaxSAT と QMF の不等式に対する 4 つの不等式の QUBO 変換手法を比較した. ここでは, 一般的に不等式を利用して定式化される最適化問題に対してこの 4 つの変換手法を適用し, 変数の数や解の得やすさを比較する.

B.1 ナップザック問題

この問題では, N 個の品物が与えられ, i 番目の品物には価値 v_i と重み w_i が与えられる. ここから, 品物の重みの総和がナップザック重みの許容量 W 以下になり, 品物の価値の総和を最大化するような品物の組合せを求める問題である [57]. この問題の目的は, 選んだ品物の価値の総和を最大化することであり, 品物 i が選ばれるとき 1 となる二値変数 $x_i \in \{0, 1\}$ を用いて次のように定式化される.

$$H_{cost}^{KP} = - \sum_{i=0}^{N-1} v_i x_i \quad (\text{B.1})$$

また, この問題の制約は選んだ品物の重みの総和がナップザック重みの許容量 W 以下になることであり, 次の不等式で定式化される.

$$\sum_{i=0}^{N-1} w_i x_i \leq W \quad (\text{B.2})$$

また, この問題をより一般化した問題として 2 次ナップザック問題がある [57]. この問題の目的は, 選んだ品物 i と j の組み合わせの価値 v_{ij} の総和を最大化するような品物の選択を求めることである.

$$H_{cost}^{2KP} = - \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} v_{ij} x_i x_j \quad (\text{B.3})$$

制約は式 (B.2) と同様である.

この2種類のナップザック問題に4.3節の4つの不等式のQUBO変換手法適用し、得られた解を評価した。まず、計算に必要な変数の総数を図B.1に示す。ただし、計算に必要な変数の数は、式(B.2)の不等式を変換する手法に依存し、どちらのナップザック問題でも等しくなるため図は1つしか示していない。この図から、One-hot>Unary>Linear>Binaryの順に変数が少なくなり、UnaryとOne-hotは $y = N$ の直線に平行に変数の数が増加していることがわかる。また、BinaryとLinearは、問題サイズが小さいときは $y = N$ の直線より変数が多くなるが、問題サイズが大きくなるとこの直線に近づくこともわかる。

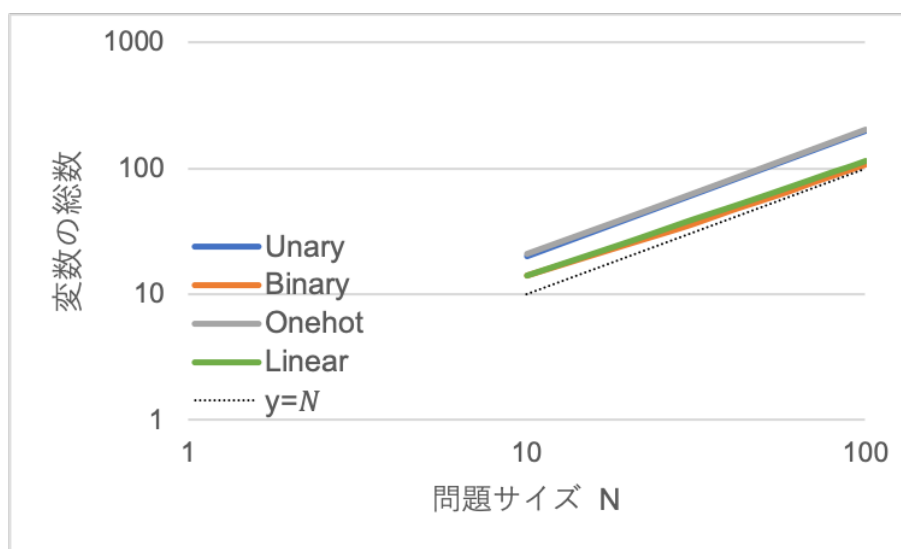


図 B.1 ナップザック問題に対する計算に必要な変数の総数

次に、実行可能解が得られた確率と得られた解のナップザック内の価値の総和（大きいほど良い）を、図B.2と図B.3に式(B.1)を利用したナップザック問題の結果、図B.4と図B.5に式(B.3)を利用した2次ナップザック問題の結果としてそれぞれ示す。まず、図B.2と図B.4からどちらのナップザック問題でも、One-hot encodingは、 $N = 60 \sim 70$ 以上で実行可能解率が下がっていくが、他の変換手法は $N = 100$ までは100%実行可能解が得られることがわかる。次に、式(B.1)を利用したナップザック問題は図B.3から、Binary>Linear>Unary>One-hotの順に価値の総和が大きい解、つまり、質の良い解が得られていることがわかる。特に、One-hot encodingでは、実行可能解率が下がり始める $N = 60$ 以上で他の変換手法よりも質の悪い解を得やすい。これらの傾向は、図B.5の式(B.3)を利用した2次ナップザック問題の結果ではより顕著に見られる。

以上の結果から、ナップザック問題では少ない変数で計算が可能な変換手法ほど質の良い解が得られると考える。

B.2 集合被覆問題

この問題では、集合 U とこの部分集合の族 σ の部分集合に対応するコスト c が与えられる。このとき、全ての要素をカバーするように部分集合の族から部分集合を選び、その際にかかるコストを最小化する問題である[46, 58]。この問題の目的はコストを最小化す

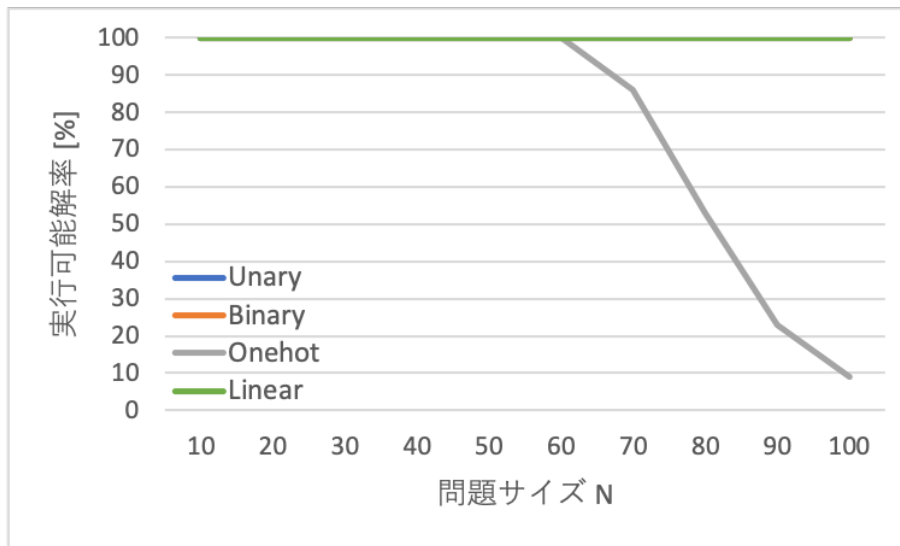


図 B.2 ナップザック問題の実行可能解が得られた確率

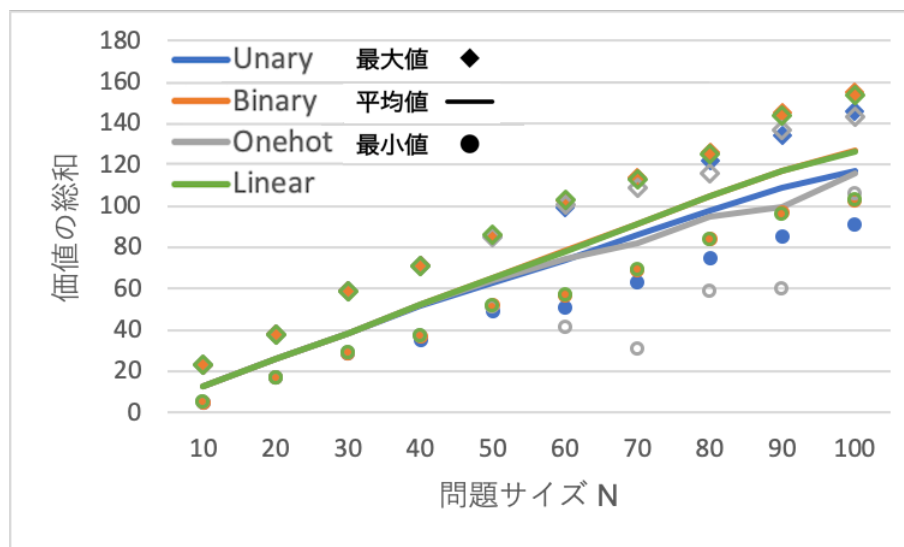


図 B.3 ナップザック問題の価値の総和

ることであり，集合 U の部分集合の族 σ_i の部分集合 j が選ばれるとき 1 となる二値変数 $x_j \in \{0, 1\}$ を用いて次のように定式化される。

$$H_{cost}^{SC} = \sum_i \sum_{j \in \sigma_i} c_j x_j \quad (B.4)$$

また，この問題の制約は全ての要素をカバーするように部分集合の族から部分集合を選ぶことであり，次の不等式で定式化される。

$$\sum_i \sum_{j \in \sigma_i} x_j \geq 1 \quad (B.5)$$

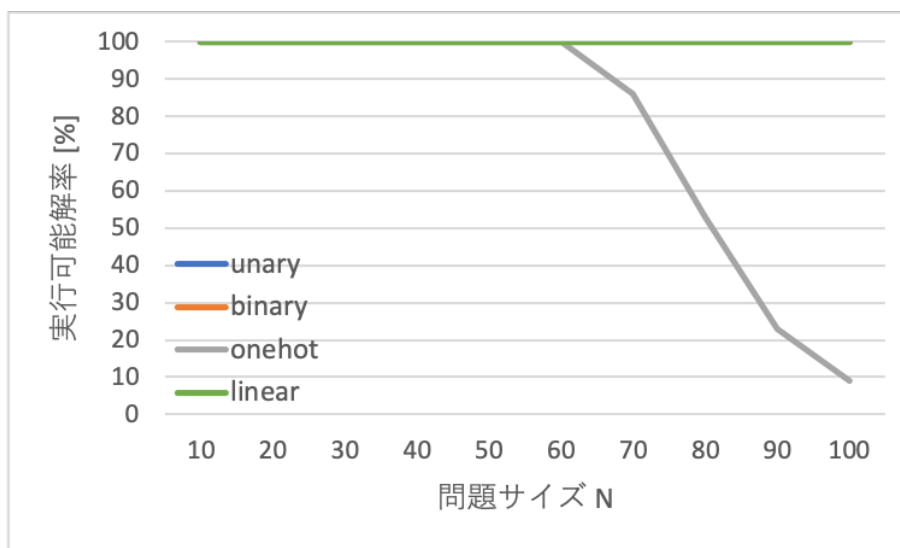


図 B.4 2次ナップザック問題の実行可能解が得られた確率

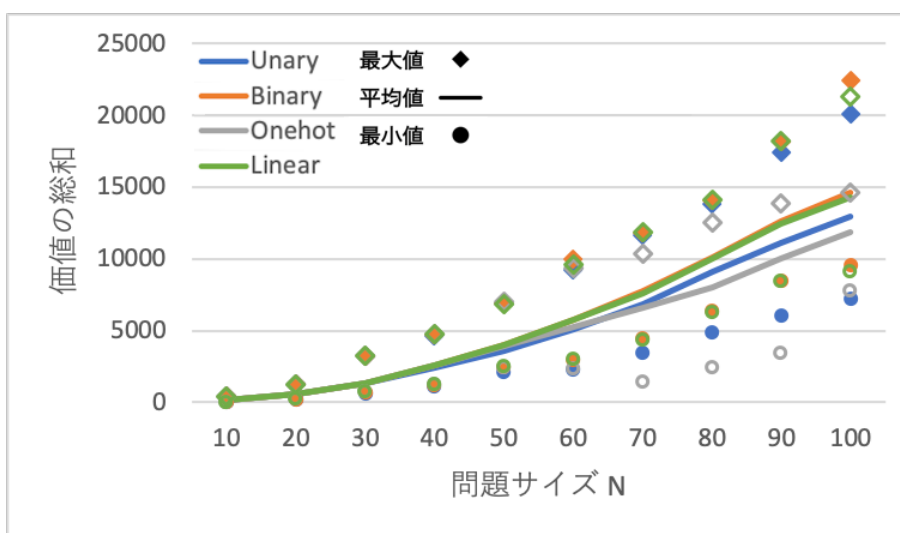


図 B.5 2次ナップザック問題の価値の総和

この問題には、 i 個の不等式で表される制約関数があるだけでなく、ナップザック問題と不等号が逆向きになるため、必要なスラック変数の数が増える。

この問題に 4.3 節の 4 つの不等式の QUBO 変換手法適用し、得られた解を評価した。まず、図 B.6 に計算に必要な変数の総数を示す。ここから、ナップザック問題と同様に、One-hot > Unary > Linear > Binary の順に変数が少なくなり、 $y = N$ の直線に対して Unary と One-hot はほぼ平行に変数の数が増加、Binary と Linear は問題サイズが大きくなるとこの直線に近づくことがわかる。また、ナップザック問題よりも変換手法間の変数の総数の差が大きくなる。これは、不等式に導入される補助変数の数がナップザック問題よりも多くなるためである。

次に、実行可能解が得られた確率を図 B.7、得られた解のコスト（小さいほど良い）を図 B.8 に示す。図 B.7 から、どの変換手法でも全ての問題サイズでほぼ 100% 実行可能解を得

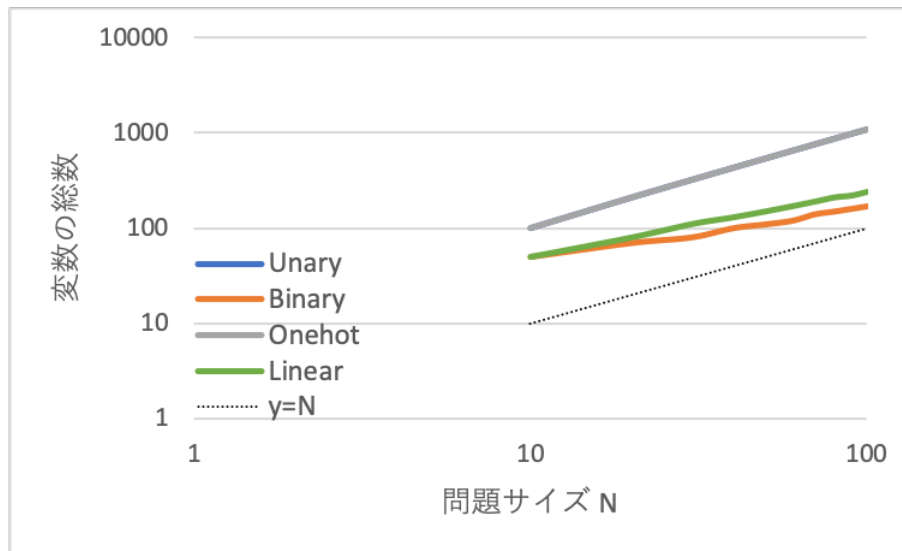


図 B.6 集合被覆問題に対する計算に必要な変数の総数

られていることがわかる。また、図 B.8 から、One-hot encoding は他の 4 手法と比較して、明らかに解の質が劣ることがわかる。One-hot encoding 以外の 4 手法の中では、得られる解の質に大きな差はないが、Unary encoding が若干他の手法よりも質の悪い解を得やすい傾向がある。

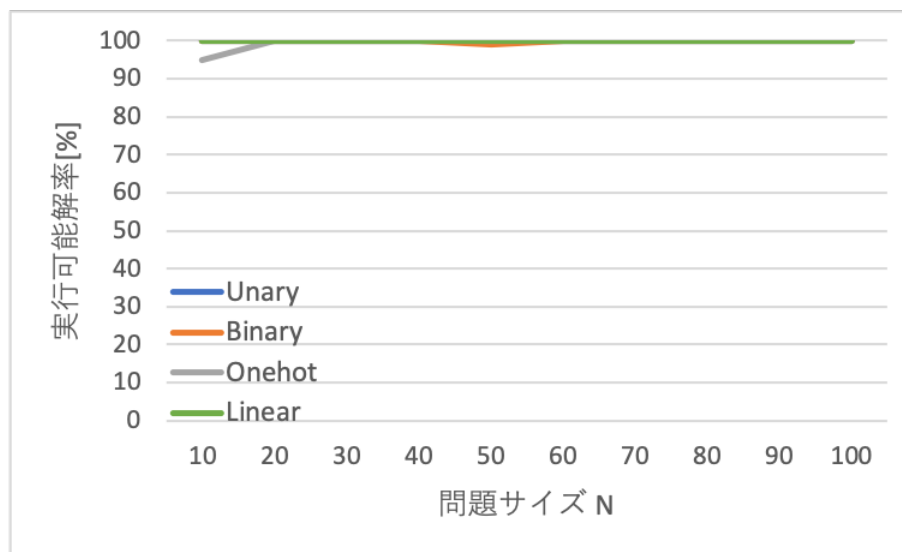


図 B.7 集合被覆問題の実行可能解が得られた確率

以上の結果から、集合被覆問題においても変数の数が少ない Binary encoding や Linear encoding が適していると考えられる。

B.3 時間制約付き経路最適化

この問題は、設定時間内に巡回可能であり、訪れる地点の満足度の総和を最大化する問題である。小見山 [59] は、地点 i から地点 j への移動を表す変数 $x_{ij} \in \{0, 1\}$ を利用して 5 つ

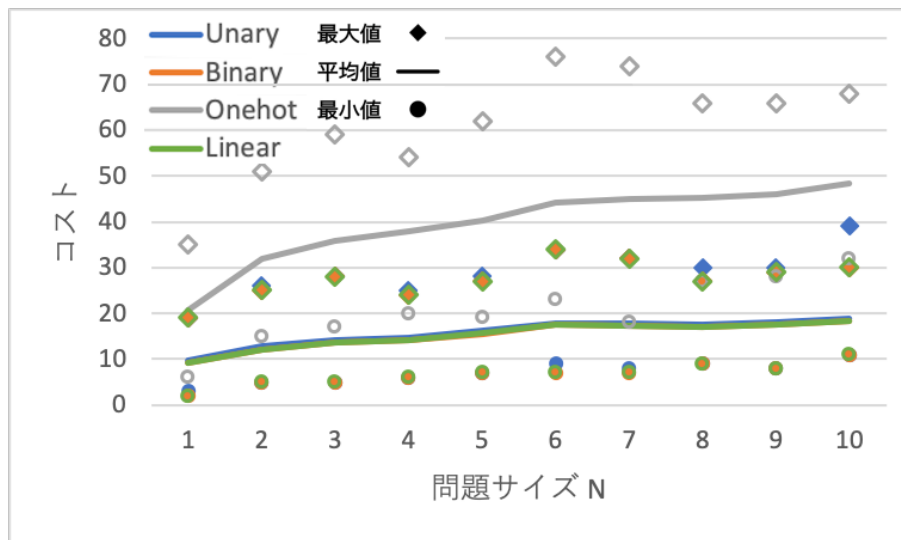


図 B.8 集合被覆問題の解のコスト

の等式制約により QUBO 定式化した（この 5 つの等式制約関数に関しては今回は説明を省略する）。しかし、これらの制約を全て満たしても主巡回路の他に副巡回路が発生することがわかった。この課題に対して、発生した副巡回路 S が次の計算の際に発生しないような不等式制約（式 (B.6)）を追加していく反復計算により、副巡回路が発生しないような解を得た。

$$\sum_{v_i \in S} \sum_{v_j \in S} x_{ij} \leq |S| - 1 \quad (\text{B.6})$$

この問題に 4.3 節の 4 つの不等式の QUBO 変換手法適用し、得られた解を評価した。今回は 5 地点の副巡回路 1 つのみで評価を行ったため、変数の総数ほどの定式化手法でもほとんど差がなかった。実行可能解が得られた確率を図 B.9 に示す。ここから、おおよそ Unary > Linear = One-hot > Binary の順に実行可能解が得られやすいことがわかる。ほとんどの変換手法で実行可能解の数が極めて少なかったため解の質の評価は省略する。

以上の結果から、時間制約付き経路最適化には、Unary encoding のような表現力が高い変換手法を適用することで実行可能解が得られやすいと考える。

以上の結果と、4.3.1 節の QMF と MaxSAT の結果から、不等式以外の制約も複数ある複雑な問題には Unary encoding などの表現力の高い変換手法、制約が少ないシンプルな問題には Binary encoding や Linear encoding などのスラック変数の数が少ない変換手法が適していると考えられる。また、One-hot encoding のような制約を増やしてしまう変換手法は、現状の QA マシンや QIA マシンでは実行可能解が得られにくく、適していないと考えられる。この結果は、不等式の QUBO 変換手法を選択するための新たな指針となる。

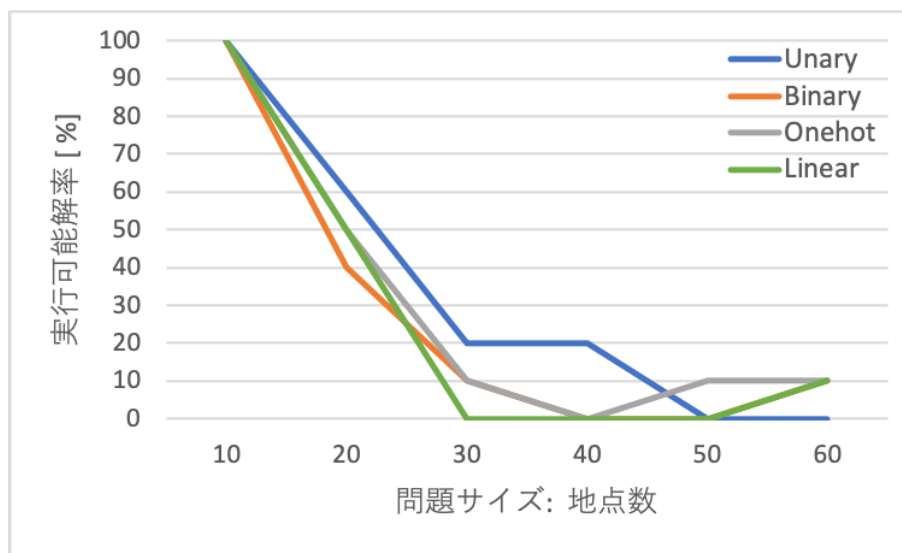


図 B.9 時間制約付き経路最適化の実行可能解が得られた確率