学外秘

# TSP 問題の QUBO モデルに関する二次項の削減方法

広島大学大学院先進理工系科学研究科　情報科学プログラム　指導教官：中野 浩嗣
M230641　劉　崇玖　所　属：コンピュータ・システム研究室

**概要**

近年，量子計算および量子アニーリングに関連する技術の発展が注目を集めている．さまざまな組合せ最適化問題は，量子アニーリングを利用することで迅速に解決可能となる．そのためには，解決したい問題を QUBO（Quadratic Unconstrained Binary Optimization）形式に変換する必要がある．巡回セールスマン問題（TSP）は，組合せ最適化問題の一例として広く知られているが，対応する QUBO モデルにおける二次項の数は，問題の規模が大きくなるに従い指数的に増加するという課題が存在する．

本研究では，ドロネー三角形分割およびボロノイ図を利用し，TSP 問題の完全グラフから最適巡回路に寄与しない辺を削除することで，対応する QUBO モデルの目的関数に含まれる二次項の数を大幅に削減する手法を提案する．実験結果では，提案手法である seg 法および nei 法による削減率はそれぞれ 68.59 % および 56.58 %を達成した．

## 1. はじめに

組合せ最適化問題とは，様々な制約条件の下で，数多くの選択肢から特定の評価関数を最大化または最小化する選択肢を見つける問題であり，一般的に NP 困難問題に分類される．問題に含まれる変数の数が増加するにつれて，解空間は指数的に拡大し，非常に難解な問題へと発展する．しかしながら，現在のコンピュータでは，組合せ最適化問題を解くのに膨大な計算時間が必要である．

通常，こうした問題に対しては多数の近似アルゴリズムが考案されており，特にメタヒューリスティックアルゴリズムが最も広く利用されている手法である [1]．遺伝的アルゴリズム，蟻コロニー最適化アルゴリズム，タブーサーチ，焼きなまし法といったアルゴリズムが代表的なメタヒューリスティックアルゴリズムとして知られている．しかし，メタヒューリスティックアルゴリズムを使用して得られる解は必ずしも最適解であることが保証されているわけではなく，最適解に近い解を導出することが目標となる．さらに，これらのアルゴリズムは汎用的であり，特定の問題に限定されずに広く適用できるが，精度の高い解を得るためには，問題に応じてアルゴリズムのパラメータを事前の知識に基づいて調整する必要がある [2]．

量子計算および量子アニーリングの急速な発展に伴い，これらの技術が組合せ最適化問題を解くための新たな手法として注目を集めている [3]．特に，量子アニーリングは量子ビットの性質を活用して，短時間で最小エネルギーを持つ解を導き出す能力を持つ．このため，量子アニーリングに関連する研究も急増している．

しかしながら，現在の量子アニーラが保有する量子ビット数はまだ限られており，解ける問題のサイズも小規模にとどまっている [4]．このため，変数の数や二次項の数を削減した QUBO モデルへの変換手法は現状の量子アニーラにおいて非常に重要な課題となっている．本研究では．TSP 問題に焦点を当て，その QUBO モデルに関する二次項の削減方法を提案する．

## 2. 組み合わせ最適化問題

### 2.1. QUBO 問題

QUBO（Quadratic Unconstrained Binary Optimization）問題とは二次形式の制約なし二値変数最適化問題であり，一般の数学式は：

$$E(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j \quad (1)$$

ここで，$x_i$ はバイナリ変数であり，その値は 0 または 1 を取る．$Q_{ii}$ は一次項の係数，$Q_{ij}$ は二次項の係数である．QUBO 問題の目標はバイナリ変数の値を適切に定め，QUBO 形式の数式を最小化または最大化することである．バイナリ変数が 4 つある QUBO 問題の例を式 2 のように挙げる．

$$E(x_1, x_2, x_3, x_4) = x_1 + 2x_2 - 3x_4 + 5x_1x_2 - 2x_2x_3 + x_1x_3 - 4x_2x_4 \quad (2)$$

ここで，$x_1, x_2, x_3, x_4$ はバイナリ変数であり，一次項と二次項はそれぞれ 3 つと四つ存在する．なお，QUBO 問題は上三角行列の形で表現することができる．行列 $Q$ の対角成分には一次項の係数 $Q_{ii}$ が配置され，上三角部分には二次項の係数 $Q_{ij}$ が配置される．図 1 は，先に示した QUBO 問題の QUBO 行列を表したものである．

バイナリ変数 $x_1, x_2, x_3, x_4 = 0, 1, 1, 1$ のとき，エネルギー関数の値は最小値であり $-7$ となる．したがって，この QUBO 問題の解は $0, 1, 1, 1$ である．

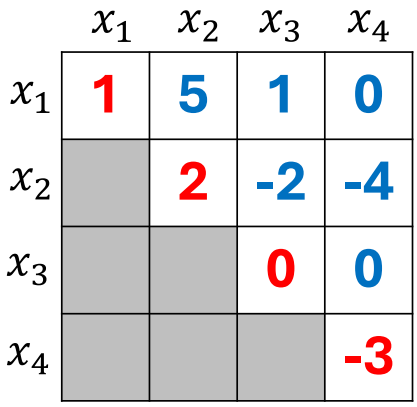


図 1: qubo 問題の行列

### 2.2. TSP 問題

TSP（巡回セールスマン問題）は，組合せ最適化問題として広く知られている [5]．TSP は町の座標や距離行列が与えられた場合に全ての町を一度ずつ訪問し出発地点に戻る巡回路の中で，総移動距離が最小となるものを求める問題である．町が 5 つ存在する TSP 問題を例に説明する．町の間の距離行列が図 2 のように与えられているとする．

| | 町1 | 町2 | 町3 | 町4 | 町5 |
|---|---|---|---|---|---|
| 町1 | 0 | 1 | 3 | 3 | 2 |
| 町2 | | 0 | 2 | 3 | 7 |
| 町3 | | | 0 | 2 | 6 |
| 町4 | | | | 0 | 3 |
| 町5 | | | | | 0 |

図 2: TSP 問題の距離行列

例として挙げたインスタンスにおいて，最適な巡回路は 1-2-3-4-5-1 となる．この巡回路における総移動距離が $1+2+2+3+2=10$ で他の全ての巡回経路と比較して最小であり，TSP 問題の解として最適である．

### 2.3. TSP 問題の QUBO モデル

通常のメタヒューリスティックアルゴリズムを用いて TSP 問題を解くことが可能であるが [6]，TSP 問題を対応する QUBO モデルに変換することで量子アニーラや QUBO ソルバーを用いて解決することもできる．TSP の QUBO モデルは，目的関数と制約条件から変換された QUBO 式で構成されており，大きく 2 つの部分に分けられる [7]．一つは巡回路の総移動距離を最小化する目的関数であり，もう一つは全ての町を一度ずつ訪れると同じタイミングで複数の町に

| | time1 | time2 | time3 | time4 |
|---|---|---|---|---|
| city1 | 0 | 1 | 0 | 0 |
| city2 | 1 | 0 | 0 | 0 |
| city3 | 0 | 0 | 0 | 1 |
| city4 | 0 | 0 | 1 | 0 |

図 3: TSP 問題の二つの制約条件

訪れることができないという制約条件を反映した項である．次に，町が 4 つある TSP 問題を例として，QUBO モデルを作成する手順を説明する．

まずは各町を訪問する順番を表すバイナリ変数 $x_{i,t}$ を式 3 のように定義する．

$$x_{i,t} = \begin{cases} 1, & \text{町 } i \text{ へ } t \text{ 番目訪れる} \\ 0, & \text{町 } i \text{ へ } t \text{ 番目訪れない} \end{cases} \tag{3}$$

したがって，町の個数が 4 である場合，必要なバイナリ変数の数は $4 \times 4 = 16$ 個となる．

#### 2.3.1. 目的関数

目的関数は式 4 で示す．

$$\sum_{i=1}^{4}\sum_{j=1}^{4}\sum_{t=1}^{4} d_{i,j}x_{i,t}x_{j,(t+1)\%4} \tag{4}$$

ここで，4 は町の個数を表し，$d_{i,j}$ は町 $i$ と $j$ の間のユークリッド距離（整数値）を表す．町 $i$ が $t$ 番目，町 $j$ が $t+1$ 番目に訪問される場合，その距離 $d_{i,j}$ が目的関数に加算される．このようにして，巡回路の各連続する町間の距離が積み重ねられ，総移動距離が目的関数の値として表現される．この距離を最小化することが TSP 問題の目標となる．

#### 2.3.2. 制約条件から変換された QUBO 式

TSP 問題には，二つの制約条件を QUBO 式に変換する必要がある．一つは，全ての町を一度ずつ訪問するという制約であり，もう一つは，同じ時間に複数の町を訪れることができないという制約である．これらの制約を反映させることで，問題全体を QUBO モデルとして表現できるようになる．

一つ目の制約条件は，図 3 の赤線で示すように，各行の変数のうち 1 つだけが 1 になり，他は全て 0 になるというものである [8]．これは各町が巡回路上で一度だけ訪問されることを意味する．式 5 で一つ目の制約条件を示す．

$$\sum_{t=1}^{4} x_{i,t} = 1 \quad (i = 1, 2, 3, 4) \tag{5}$$

ペナルティー法を用いて移項と二乗すると対応する QUBO モデルが式 6 になる．

$$\sum_{i=1}^{4}\left(\sum_{t=1}^{4}x_{i,t}-1\right)^2 \tag{6}$$

二つ目の制約条件は，同じタイミングに複数の町を訪れることができないというものであり．図 3 の青線で示すように，各列の変数のうち 1 つだけが 1 になり，他は全て 0 になるというものである [8]．この条件により，同一のタイミングで複数の町を訪問することが防がれる．式 7 で二つ目の制約条件を示す．

$$\sum_{i=1}^{4}x_{i,t}=1 \quad (t=1,2,3,4) \tag{7}$$

ペナルティー法を用いて移項と二乗すると対応する QUBO モデルが式 8 になる

$$\sum_{t=1}^{4}\left(\sum_{i=1}^{4}x_{i,t}-1\right)^2 \tag{8}$$

### 2.3.3. 全体の QUBO モデル

問題全体の QUBO モデルは式 4 と 6 と 8 から構成されている．

$$\lambda\left(\sum_{i=1}^{4}\left(\sum_{t=1}^{4}x_{i,t}-1\right)^2+\sum_{t=1}^{4}\left(\sum_{i=1}^{4}x_{i,t}-1\right)^2\right) \tag{9}$$

ここで，$\lambda$ はペナルティー係数であり，通常は距離行列の最大値に設定される．制約条件を違反した場合，ペナルティーとして式の値を増加させ，制約を守る解が選択されるようにする役割を果たす．全体のモデルを展開すると図 4 になる．ここで，赤色で表示されている部分は目的関数における二次項を示していて，このインスタンスには目的関数の二次項数は 48 である．

| | x1,1 | x1,2 | x1,3 | x1,4 | x2,1 | x2,2 | x2,3 | x2,4 | x3,1 | x3,2 | x3,3 | x3,4 | x4,1 | x4,2 | x4,3 | x4,4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x1,1 | -2λ | 2λ | 2λ | 2λ | 2λ | d1,2 | | d1,2 | 2λ | d1,3 | | d1,3 | 2λ | d1,4 | | d1,4 |
| x1,2 | | -2λ | 2λ | 2λ | d1,2 | 2λ | d1,2 | | d1,3 | 2λ | d1,3 | | d1,4 | 2λ | d1,4 | |
| x1,3 | | | -2λ | 2λ | | d1,2 | 2λ | d1,2 | | d1,3 | 2λ | d1,3 | | d1,4 | 2λ | d1,4 |
| x1,4 | | | | -2λ | d1,2 | | d1,2 | 2λ | d1,3 | | d1,3 | 2λ | d1,4 | | d1,4 | 2λ |
| x2,1 | | | | | -2λ | 2λ | 2λ | 2λ | d2,3 | | d2,3 | 2λ | d2,4 | | d2,4 | |
| x2,2 | | | | | | -2λ | 2λ | 2λ | d2,3 | 2λ | d2,3 | | d2,4 | 2λ | d2,4 | |
| x2,3 | | | | | | | -2λ | 2λ | | d2,3 | 2λ | d2,3 | | d2,4 | 2λ | d2,4 |
| x2,4 | | | | | | | | -2λ | d2,3 | | d2,3 | 2λ | d2,4 | | d2,4 | 2λ |
| x3,1 | | | | | | | | | -2λ | 2λ | 2λ | 2λ | 2λ | d3,4 | | d3,4 |
| x3,2 | | | | | | | | | | -2λ | 2λ | 2λ | d3,4 | 2λ | d3,4 | |
| x3,3 | | | | | | | | | | | -2λ | 2λ | | d3,4 | 2λ | d3,4 |
| x3,4 | | | | | | | | | | | | -2λ | d3,4 | | d3,4 | 2λ |
| x4,1 | | | | | | | | | | | | | -2λ | 2λ | 2λ | 2λ |
| x4,2 | | | | | | | | | | | | | | -2λ | 2λ | 2λ |
| x4,3 | | | | | | | | | | | | | | | -2λ | 2λ |
| x4,4 | | | | | | | | | | | | | | | | -2λ |

図 4: モデルの QUBO 行列

町 $n$ が個あるインスタンスに対して，目的関数の二次項数は $n^2(n-1)$ である．したがて，インスタンスのサイズが大きくなるにつれて，目的関数の二次項数が爆発的に増加するという問題が生じる．

本研究では，新しいアイデアとしてドロネー三角形分割およびボロノイ図を利用し，元の TSP 問題の完全グラフから最適巡回路に寄与しない辺を削除することで，目的関数の二次項数を削減する手法を提案している．

### 2.3.4. ボロノイ図とドロネー三角分割

ボロノイ図（Voronoi Diagram）とは，平面上に配置された複数の点（母点）に対して，各点から最も近い領域を決定する方法である．具体的には，各母点からの距離が最小となる点の集合が，その母点に対応するボロノイ領域を形成する．これにより，平面が複数のボロノイ領域に分割される [9]．図 5 はある TSP インスタンスに基づいて生成されたボロノイ図，複数のボロノイ領域から構成されている．青い点は母点（TSP 問題の町），オレンジ色の点はボロノイ頂点，黒い辺はボロノイ辺である．

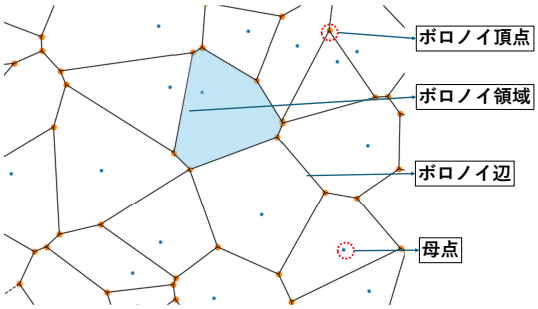


図 5: ボロノイ図

ドロネー三角分割（Delaunay Triangulation）は，与えられた母点集合を三角形に分割する手法で，ボロノイ図と密接に関連している．具体的には，ドロネー三角分割は，ボロノイ図の隣接する領域の母点を結んで得られる三角形から構成される．このように，ボロノイ図とドロネー三角形分割は双対の関係にあり，ボロノイ図の頂点はドロネー三角形の外接円の中心に対応する．この関係性により，ドロネー三角形分割とボロノイ図は計算幾何学やグラフィックス，地理情報システムなど，さまざまな分野で応用されている [10]．図 6 にはボロノイ図から得られたドロネー三角形分割が表示されている．この三角形分割は，町間の空間的関係を反映しており，TSP 問題における最適な巡回路の推定に役立つ．

ドロネー三角形分割の辺は，必ずしも最適巡回路を含むわけではないため，他の辺を加える必要がある．本研究では，ボロノイ図に基づく辺を加える手法を提案し，元の TSP 問題の完全グラフから最適巡回路に寄与しない不要な辺を削除し，効率的な QUBO モデルを構築することを目指す．

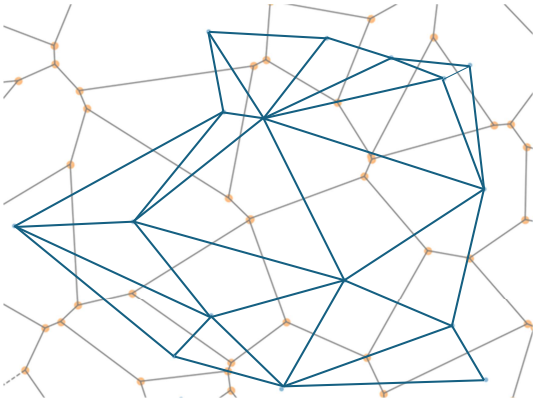

図 6: ドロネー三角分割

## 3. 提案手法

ドロネー三角形分割の辺は，TSP 問題の最適巡回路になる可能性が高いが，全ての最適巡回路が必ずしもドロネー三角形分割に含まれるわけではないという問題がある．本研究では，最適巡回路が新しいグラフに含まれるようにするための 2 つの新しい手法を提案する．これらの手法は，既存のドロネー三角形分割に新しい辺を加えることで，新しいグラフを構築するものである．

### 3.1. nei 方法

ドロネー三角形分割を構築する一つの手法として，ボロノイ図で隣接する 2 つの領域に属する母点を繋げることでドロネー三角形分割が得られる．提案する nei 手法では，この隣接関係を一度だけでなく，最大 3 回まで拡張する．つまり，1 つのボロノイ領域に対して，隣の隣，さらにはその隣の領域まで母点を繋げる手法である．図 7 に nei 手法の例を示す．

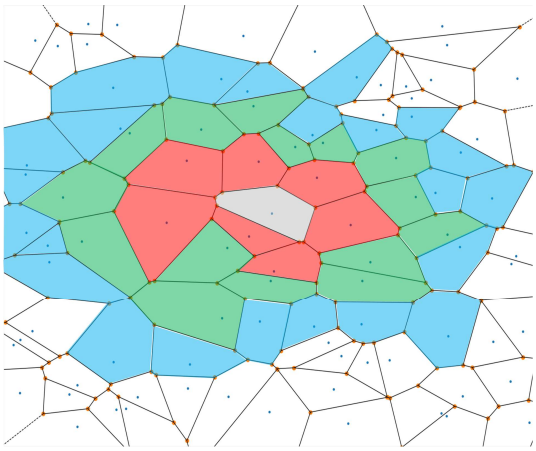

図 7: nei 方法

図 7 に示すように，赤色，緑色，青色の領域はそれぞれ隣接関係を 1 回，2 回，3 回拡張した結果を表している．提案する nei 手法では，中央の灰色領域の母点を他の 3 色の領域に属する母点と繋げる．この手法により，nei グラフに含まれる辺の数が増加し，最適巡回路が含まれる可能性が高まる．

### 3.2. seg 方法

図 8 に示すように，赤色のボロノイ辺に対応する両端の赤色の点線で囲まれた母点を繋げる．さらに，緑色の 2 つ連続するボロノイ辺，および青色の 3 つ連続するボロノイ辺に対応する両端の母点も繋げる．この手法により，seg グラフに含まれる辺の数が増え，nei グラフと同様に，最適巡回路が含まれる可能性が高くなる．

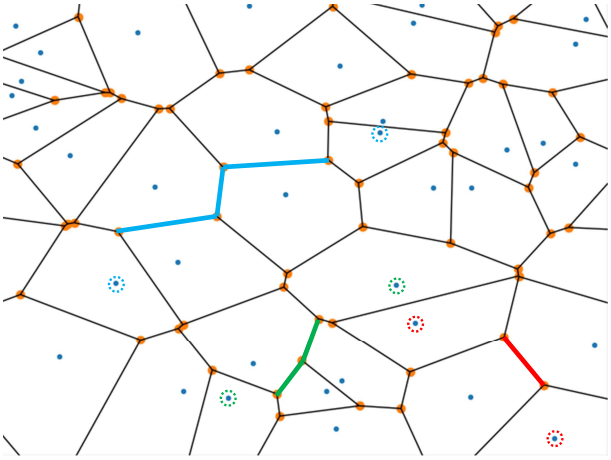

図 8: seg 方法

## 4. 実験

実験で使用するインスタンスは，まずランダムに町の座標を生成し，その後，四捨五入した距離行列を計算して作成する．この距離行列は上三角行列の形で TSP 問題専用ソルバーである LKH[11] に入力される．また，1 つのインスタンスに対して，同じパラメータで 10 回繰り返し解を求め，その中で最短の巡回路をインスタンスの最適解とする．実験に用いるインスタンスのサイズ（町の個数）は 5 から 200 までで，総計 196 個のインスタンスを使用する．

### 4.1. 辺の個数に関する実験

ドロネー三角形分割，seg 方法で計算されたグラフ，nei 方法で計算されたグラフ，そして完全グラフのそれぞれにおける辺の個数を統計した結果は，図 9 に示されている．

図 9 が示すように，辺の個数が最も多いのは当然ながら完全グラフであり，次に多いのは nei 方法で計算されたグラフと seg 方法で計算されたグラフである．最も少ない辺を持つのはドロネー三角形分割である．インスタンスのサイズが小さい場合，どの計算方法でも得られたグラフの辺の個数はほぼ同じであるが，サイズが大きくなるにつれて，提案された seg 方法と nei 方法で得られたグラフは，完全グラフと比較して大幅に辺を削除できることが分かる．完全グラフからより多くの辺を削除することで，QUBO モデルに

おける目的関数の二次項数を減少させる効果が得られる．

### 4.2. 提案された seg グラフと nei グラフには最適巡回路が含まれるか

上述のように，各インスタンスの最適巡回路は TSP 問題専用ソルバーである LKH を用いて得られた．一つのインスタンスに複数の最適巡回路が存在する可能性はあるが，LKH で TSP 問題を解いた場合には，一つの最適巡回路のみが得られる．その最適巡回路が提案した seg グラフおよび nei グラフに含まれているかどうかを確認した結果，全てのインスタンスにおいて，提案された seg 方法および nei 方法で得られたグラフ内に LKH で得られた最適巡回路が含まれていることが確認された．したがって，元の完全グラフに基づく TSP 問題と，提案された方法で得られたより少ない辺を持つグラフに基づく TSP 問題は同等であり，制約された TSP 問題の方が計算が容易になることが期待される．

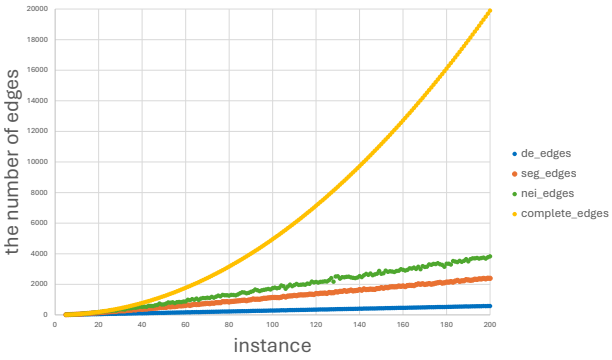


図 9: 異なるグラフの辺の個数

### 4.3. LKH で制限されたグラフ（seg グラフと nei グラフ）の TSP 問題を解く

一つのインスタンスに対して，複数の最適巡回路が存在する可能性があるが，LKH では一つの最適巡回路のみが出力される．LKH を用いて制限されたグラフの TSP 問題を解くことで，他の最適巡回路が存在するかどうかを確認できる．具体的な手順としては，LKH に入力する距離行列において，制限されたグラフに存在しない辺（完全グラフと比較した場合）の距離を，元の距離行列の最大値に書き換える．この操作により，その辺が必ず最適巡回路に含まれないことを意味し，制限された TSP 問題の距離行列を作成できる．

この制限された距離行列を LKH に入力し，同じパラメータで問題を解いた後，得られた解の総距離と巡回路を比較することで，複数の最適巡回路が存在するかどうかを確認できる．結果として，全てのインスタンスにおいて，制限なし（完全グラフ）の TSP 問題の最適巡回路の総距離と，制限された（seg グラフまたは nei グラフ）TSP 問題の最適巡回路の総距離が一致することが確認された．また，対応する 2 つの巡回路の町の順番を確認したところ，多数のインスタンスで，制限された TSP 問題の最適巡回路と，制限なしの TSP 問題の最適巡回路が異なる巡回路であることが確認できた．したがって，多くのインスタンスで複数の最適巡回路が存在することがわかるが，提案された手法で作成されたグラフには，少なくとも 1 つの最適巡回路が含まれている．

## 5. 目的関数の二次項数を削減する

### 5.1. 二次項数を削減する手順

第 4 章で行われた実験のもとで制限されたグラフに存在しない辺は最適巡回路にならないので元の TSP 問題の QUBO モデルを作るときそれらの考慮をしなくても良いということで目的関数の二次項数を削減できる．具体的な手順は図 10 で町五つあるインスタンスで説明する．制限されたグラフの距離行列では，完全グラフと比較して存在しない辺の部分を，元の完全グラフに基づく距離行列の最大値に書き換える．その後，書き換えた距離行列の全ての要素（対角成分を除く）から距離行列の最大値を引く．この操作により，全要素から同一の値を引いても問題の本質は変わらないため，問題に影響を与えない．これにより，制限されたグラフに存在しない辺の部分は 0 となり，QUBO モデルを作成する際に，それに関連する二次項を削減できる．

### 5.2. 実験結果

二つ提案手法の削減率を評価するために，本研究では削減率を式 10 のように定義する．

$$\text{削減率} = \frac{\text{削減できる二次項数}}{\text{元の目的関数の二次項数}} \times 100\% \quad (10)$$

図 11 に示すように，インスタンスのサイズが大きくなるにつれて，提案した 2 つの手法における二次項数の削減率も向上している．特に，seg グラフは nei グラフよりも辺の個数が少ないため，seg 方法での削減率は nei 方法よりも高い．

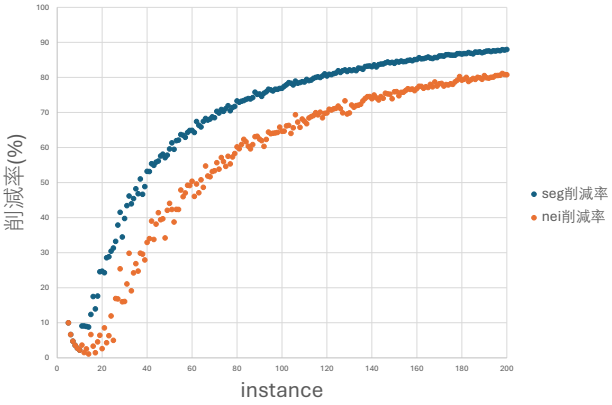


図 11: seg 方法と nei 方法の削減率

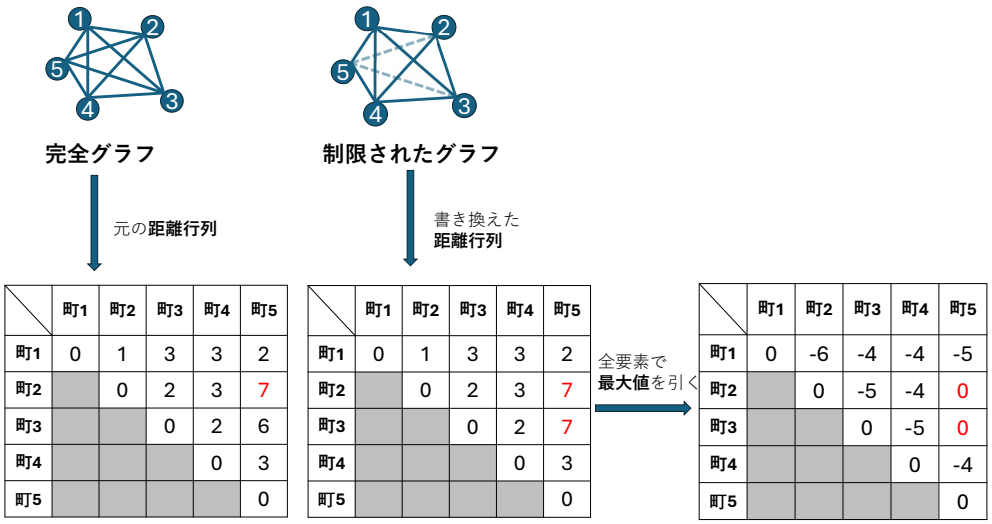


図 10: 目的関数の二次項を削減方法

総計 196 個のインスタンスを統計した結果，seg 方法と nei 方法の平均削減率はそれぞれ 68.59 %と 56.68 %であった．

### 5.3. まとめ

本研究では，TSP 問題の QUBO モデルに注目し，元の完全グラフから最適巡回路にならない辺を削除する手法を提案した．これにより，対応する QUBO モデルを作成する際に，制限されたグラフに存在しない辺を考慮する必要がなくなるというアイデアである．このアイデアに基づき，2 つの提案手法における目的関数の二次項数削減率を実験で評価した．結果として，総計 196 個のインスタンスにおいて，seg 方法と nei 方法の平均削減率はそれぞれ 68.59 %と 56.68 %であった．

### Reference


[1] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, “Chapter 10 - metaheuristic algorithms: A comprehensive review,” in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*. Academic Press, 2018, pp. 185–231.

[2] M. D. García, M. Ayodele, and A. Moraglio, “Exact and sequential penalty weights in quadratic unconstrained binary optimisation with a digital annealer.” Association for Computing Machinery, 2022.

[3] F. Glover, G. Kochenberger, and Y. Du, “A tutorial on formulating and using qubo models,” *arXiv preprint arXiv:1811.11538*, 2018.

[4] J. Cai, W. G. Macready, and A. Roy, “A practical heuristic for finding graph minors,” *arXiv preprint arXiv:1406.2741*, 2014.

[5] E. Osaba, X.-S. Yang, and J. Del Ser, “Chapter 9 - traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics,” in *Nature-Inspired Computation and Swarm Intelligence*, X.-S. Yang, Ed. Academic Press, 2020, pp. 135–164. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128197141000208

[6] D. Gupta, “Solving tsp using various metaheuristic algorithms,” *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, vol. 1, no. 2, pp. 22–26, 2013.

[7] A. Lucas, “Ising formulations of many np problems,” *ArXiv*, vol. abs/1302.5843, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:10554455

[8] S. Okada, M. Ohzeki, and S. Taguchi, “Efficient partition of integer optimization problems with one-hot encoding,” *Scientific reports*, vol. 9, no. 1, p. 13036, 2019.

[9] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Comput. Surv.*, vol. 23, p. 345–405, sep 1991.

[10] S. Fortune, “Voronoi diagrams and delaunay triangulations,” in *Handbook of discrete and computational geometry*, 2017, pp. 705–721.

[11] M. A. Ismail, S. H. Mirza, and T. Altaf, “A parallel and concurrent implementation of linkernighan heuristic (lkh-2) for solving traveling salesman problem for multi-core processors using spc 3 programming model,” *IJACSA Editorial*, 2011.