

修士論文要旨

Summary of the Master's Thesis

学生番号 Student ID Number	M230641	プログラム Program	情報科学プログラム
氏名 Name	LIU CHONGJIU	主指導教員 Supervisor	中野 浩嗣
論文題目 Thesis Title	TSP 問題の QUBO モデルに関する二次項の削減方法		

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 モデルの二次項の数を大幅に削減する手法を提案する。

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。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 [1]。

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 [2]。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 [3]。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：

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

ここで、 x_i はバイナリ変数であり、 n はその個数である。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。

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 [4]。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。町の間距離行列が図 1 のように与えられているとする。

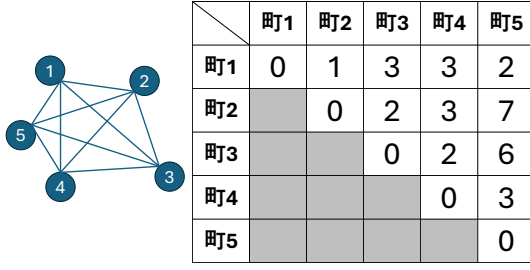


図 1: TSP 問題の距離行列

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

TSP 問題に対応する 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 [5]。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 2 のように定義する。

$$x_{i,t} = \begin{cases} 1, & \text{町 } i \text{ が } t \text{ 番目に訪れる} \\ 0, & \text{町 } i \text{ が } t \text{ 番目に訪れない} \end{cases} \quad (2)$$

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 3 で示す。

$$\sum_{i=1}^4 \sum_{j=1}^4 \sum_{t=1}^4 d_{i,j} x_{i,t} x_{j,(t+1)\%4} \quad (3)$$

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。一つ目は、全ての町を一度ずつ訪問するという制約であり、二つ目は、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。

	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 2: TSP 問題の二つの制約条件

一つ目の制約条件は、図 2 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。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 4 で一つ目の制約条件を示す。

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

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 5 になる。

$$\sum_{i=1}^4 \left(\sum_{t=1}^4 x_{i,t} - 1 \right)^2 \quad (5)$$

二つ目の制約条件は、図 2 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。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 6 で二つ目の制約条件を示す。

$$\sum_{i=1}^4 x_{i,t} = 1 \quad (t = 1, 2, 3, 4) \quad (6)$$

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 7 になる

$$\sum_{t=1}^4 \left(\sum_{i=1}^4 x_{i,t} - 1 \right)^2 \quad (7)$$

2.3.3. 全体の QUBO モデル

問題全体の QUBO モデルは式 3 と 5 と 7 から構成されていて、式 8 で示される。

$$\sum_{i=1}^4 \sum_{j=1}^4 \sum_{t=1}^4 d_{i,j} x_{i,t} x_{j,(t+1)\%4} + \lambda \left(\sum_{i=1}^4 \left(\sum_{t=1}^4 x_{i,t} - 1 \right)^2 + \sum_{t=1}^4 \left(\sum_{i=1}^4 x_{i,t} - 1 \right)^2 \right) \quad (8)$$

ここで、 λ 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。町 n が個あるインスタンスに対して、モデルの二次項数は $2n^2(n-1)$ 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。

3. 量子アニーラのトポロジーと埋め込み

3.1. 三つのトポロジー

現在、D-Wave 社が開発した量子アニーラの QPU には、三つの異なるトポロジーが存在する。これらは D-Wave 2000Q で使用されるキメラ (Chimera), Advantage で使用されるペガサス (Pegasus), 次世代のゼファー (Zephyr) という名前で知られており、それぞれが独自の接続構造を持っている。

キメラトポロジーには、縦 16 セル、横 16 セルの配置で、合計 256 個のキメラセルが搭載されている。全体で 2048 個の量子ビットが QPU 上に配置されている。

ペガサスはキメラの拡張版であり、より多くの量子ビットとより高密度の接続を特徴としている。この新しいトポロジーによって、より複雑な問題を効率的に解く能力が向上した。ペガサスが持っている量子ビットの個数も 5640 個に達した。

ゼファーは、キメラとペガサスのトポロジーをさらに進化させたものであり、量子ビット間の接続性と効率を向上させることを目的としている。ゼファートポロジーが持っている量子ビットの個数も 7440、カプラーも 71736 個に達した。

3.2. マイナー埋め込み

マイナー埋め込みは、一つのグラフが別のグラフのサブグラフとして存在することを示すためのプロセスである。これは、QPU の物理的制約内で計算を実行するために必要な手順であり、特に D-Wave の量子アニーリング技術に対して非常に重要なプロセスでもある。マイナー埋め込みを利用することで、理論上のモデルを物理的な量子アニーラ上で効果的に実装でき、最適化問題を解決できるようになる。

$$Q(x_1, x_2, x_3, x_4) = x_1 + 2x_2 - 3x_4 + 5x_1x_2 - 2x_2x_3 + x_1x_3 - 4x_2x_4 \quad (9)$$

例えば、式 9 で示される QUBO 問題を Gest グラフで表現できて、図 3 のようにマイナー埋め込みアルゴリズムで Host グラフのキメラトポロジーにマイナー埋め込みできる。

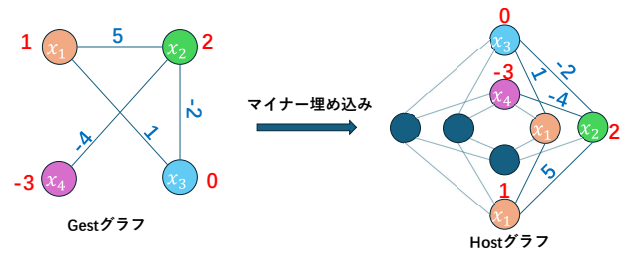


図 3: マイナー埋め込み

4. 提案手法

4.1. 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 [6]。図 4 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。

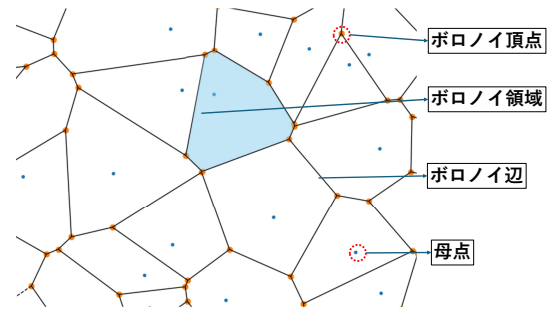


図 4: ボロノイ図

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分割は双対の関係になり、この関係性により、ドロネー三角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 [7]。図 5 にはボロノイ図から得られたドロネー三角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 [8]、含まれるようにするために他の辺を加える必要がある。

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 つの新しいグラフを構築手法を

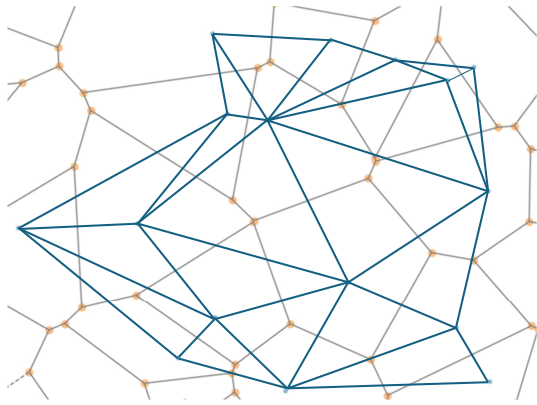


図 5: ドロネー三角形分割

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．

4.2. Nei 方法

元のドロネー三角形分割を構築する一つの手法として，ポ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つのポ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6に Nei 手法の例を示す．

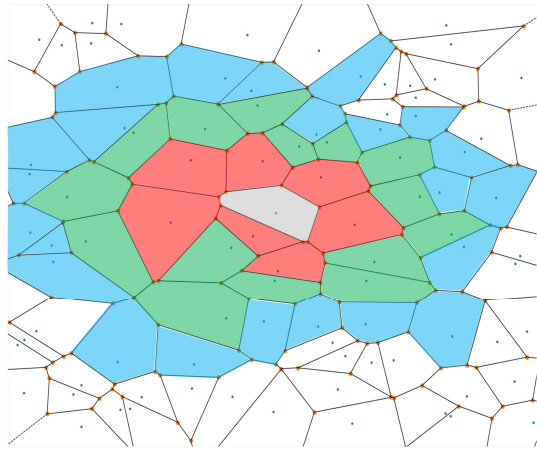


図 6: Nei 方法

図6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．

4.3. Seg 方法

図7に示すように，赤色のポ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つ連続するポロノイ辺，および青色の3つ連続するポロノイ辺に対応する両端の母点も繋げる．こ

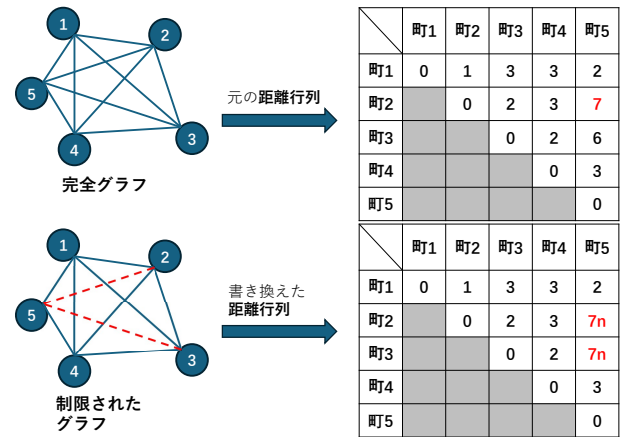


図 8: 制限された TSP 問題の距離行列

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．

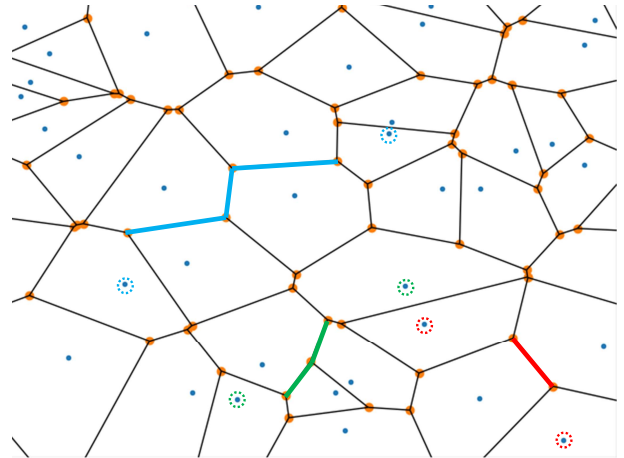


図 7: Seg 方法

5. 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[9] と Concorde[10] に入力されて最適解が得られる．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．全てのインスタンスに提案された Seg グラフと Nei グラフでは，全ての最適巡回路が含まれていることも確認されている．

5.1. 制限された TSP 問題をソルバーで解く

完全グラフから特定の辺を除外した制限された TSP 問題（ドロネー三角形分割，Nei グラフと Seg グラフそれぞれに基づく TSP 問題）をソルバーで解いて解を比較する．

図8は制限された TSP 問題の距離行列を構成する流れをサイズが5のインスタンスで説明する図で

ある．制限されたグラフに存在しない辺のところでは元の距離行列の最大値の n 倍で書き換えることで制限された TSP 問題の距離行列が得られる．ここで、 $n = 1, 5, 10$ 三つの値を設定して、二つ異なるソルバー LKH と Concorde でそれぞれ実験した．

LKH と Concorde ソルバーを用いた実験結果から、最大距離の倍数 n がいかなる値であっても、制限された TSP 問題においてグラフに存在しない辺が選ばれないことが確認された．なお、Seg グラフおよび Nei グラフに基づく TSP 問題の解のエネルギーは常に最適解のエネルギーと一致している．これにより、通常の完全グラフに基づく TSP 問題と Seg グラフまたは Nei グラフに基づく TSP 問題を解くことは本質的に同様のアプローチであると言えるでしょう．

5.2. グラフの辺の個数

ドロネー三角形分割、Seg グラフ、Nei 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．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．

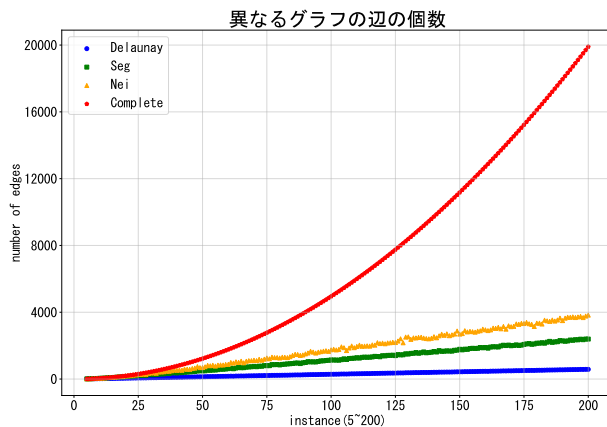


図 9: 異なるグラフの辺の個数

5.3. 目的関数の二次項数を削減する

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??で示されている．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

Table 1: 拡大されたトポロジーの量子ビット数		
トポロジー	現在の qbit 数	数式
キメラ	2048 ($m = 16$)	$8m^2$
ペガサス	5640 ($m = 16$)	$8(3m - 1)(m - 1)$
ゼファー	7440 ($m = 15$)	$32m^2 + 16m$

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．

二つ提案手法の削減率を評価するために、完全グラフに基づく TSP 問題の二次項数は X で、制限されたグラフに基づく TSP 問題の二次項数は Y で表して、削減率 (Reduction Rate) を式 10 のように定義する．

$$\text{削減率} = \frac{X - Y}{X} \times 100\% \quad (10)$$

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．

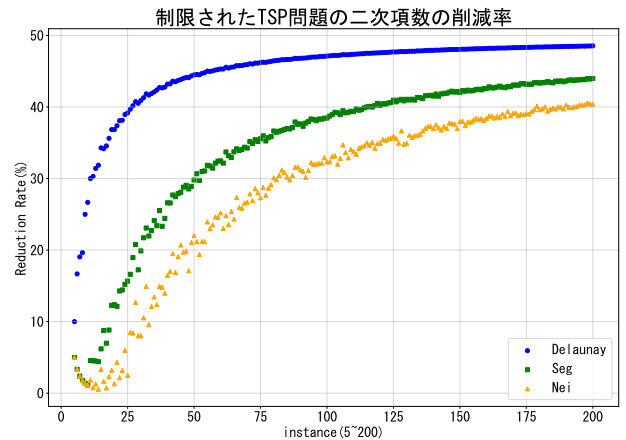


図 11: seg 方法と nei 方法の削減率

総計 196 個のインスタンスを統計した結果、三角形分割、Seg グラフ、Nei 方法の平均削減率はそれぞれ 44.6 %、34.3 %、28.3 %であった．

5.4. 異なるグラフでのマイナー埋め込み

現在のトポロジーの量子ビット数がまだ少ないため、実験で使用されるインスタンスを埋め込みできるように、現在のトポロジーのサイズが段々拡大して、埋め込み実験をする．各トポロジーで拡大されて後、使用できる量子ビット数が表のように表 1 している．

ここで、 m をパラメータとして使用される．この拡大されたトポロジーを使用することで、より大きな

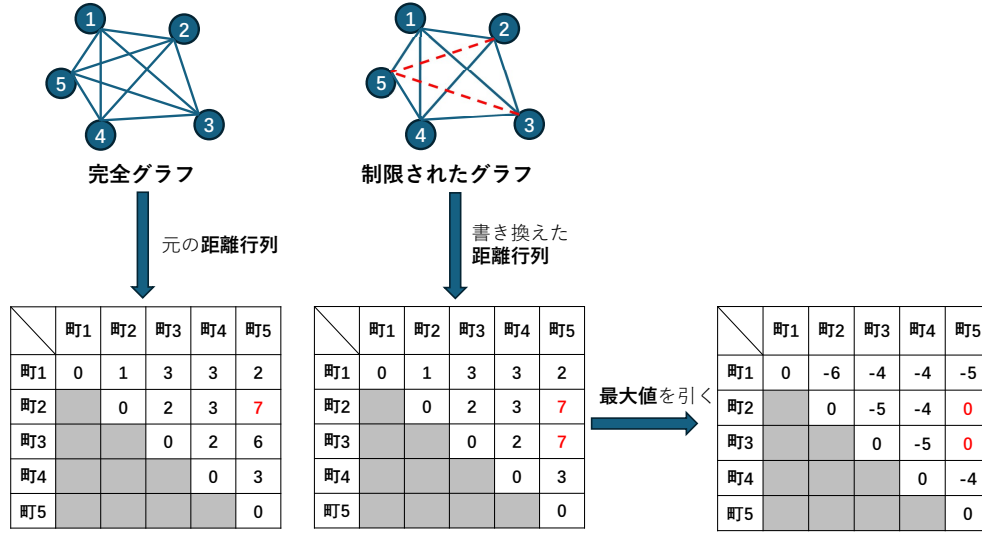


図 10: 目的関数の二次項を削減方法

インスタンスの埋め込み実験が可能となる。実験では、一つのインスタンスに対して、合計 10 回のマイナー埋め込み試行を行う。埋め込みが成功した場合、同じサイズのトポロジーでインスタンスのサイズを 1 増加させ、再度埋め込みを試みる。一方で、埋め込みが一度も成功しない場合は、インスタンスのサイズを変更せずに、対応するトポロジーのパラメータ m を 1 増加させて、さらなる埋め込み試行を行う。これにより、各サイズのトポロジーのパフォーマンス限界と拡張可能性を評価することが可能となる。

結果として、ドロネー三角形分割は最大のサイズの問題を埋め込みできて、完全グラフよりも最大 4 サイズ大きいインスタンスを、提案された Seg と Nei グラフは 1 サイズ大きいインスタンスを埋め込むことができた。

6. まとめ

本研究では、TSP 問題に注目し、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って、モデルの二次項数を削減できる。なお、マイナー埋め込みの結果として、同じサイズのトポロジーでより大きいサイズの問題を埋め込みできるようになった。

Reference

- [1] 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.
- [2] F. Glover, G. Kochenberger, and Y. Du, “A tutorial on formulating and using QUBO models,” *arXiv preprint arXiv:1811.11538*, 2018.
- [3] J. Cai, W. G. Macready, and A. Roy, “A practical heuristic for finding graph minors,” *arXiv preprint arXiv:1406.2741*, 2014.
- [4] 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*, 2020, pp. 135–164.
- [5] A. Lucas, “Ising formulations of many NP problems,” *ArXiv*, vol. abs/1302.5843, 2013.
- [6] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Comput. Surv.*, vol. 23, p. 345–405, sep 1991.
- [7] S. Fortune, “Voronoi diagrams and Delaunay triangulations,” in *Handbook of discrete and computational geometry*, 2017, pp. 705–721.
- [8] M. B. Dillencourt, “Traveling salesman cycles are not always subgraphs of Delaunay triangulations or of minimum weight triangulations,” *Information Processing Letters*, vol. 24, no. 5, pp. 339–342, 1987.
- [9] 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.
- [10] D. L. Applegate, R. E. Bixby, V. Chvátal, W. Cook, D. G. Espinoza, M. Goycoolea, and K. Helsgaun, “Certification of an optimal tsp tour through 85,900 cities,” *Operations Research Letters*, vol. 37, no. 1, pp. 11–15, 2009.